This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

This Page Blank (uspto)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 030 457 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
23.08.2000 Bulletin 2000/34

(51) Int. Cl.[7]: **H03M 13/29**, H03M 13/39

(21) Application number: **00200557.7**

(22) Date of filing: **18.02.2000**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **18.02.1999 US 120738 P**
02.09.1999 US 152514 P

(71) Applicant:
INTERUNIVERSITAIR MICROELEKTRONICA CENTRUM VZW
3001 Leuven (BE)

(72) Inventors:
- **Glese, Jochen Uwe**
  48455 Bad Bentheim (DE)
- **Schurgers, Curt**
  2370 Arendonk (BE)
- **Van der Perre, Liesbet**
  3001 Leuven (BE)
- **Gyselinckx, Bert**
  3001 Leuven (BE)
- **Catthoor, Francky**
  9140 Temse (BE)
- **Engels, Marc**
  3018 Wijgmaal (BE)

(74) Representative:
Bird, William Edward et al
Bird Goen & Co.,
Vilvoordsebaan 92
3020 Winksele (BE)

(54) Methods and system architectures for turbo decoding

(57) Methods and architectures for turbo decoding are presented. Said methods and architectures are such that low energy consumption is obtained with reduced memory requirements. Moreover said methods show improved performance with respect to latency. This is achieved by re-organising the basic decoding algorithm to optimise processing by optimizing a cost function incorporating at least one performance criteria such a energy consumption, latency and memory requirements. The optimisation may invomve the size of the memory for partially storage of computed state metrics; the size of the memory for partially storage of recomputed state metrics; the length of the sequence of valid metrics divided by the length of the sequence of invalid metrics; the number of compute and determining steps; the amount of sequentially executed valid metric calculations per compute and determining step.

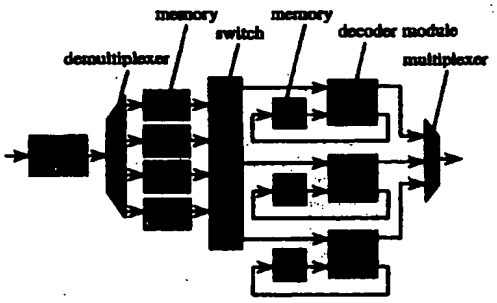


Figure 6

EP 1 030 457 A2

Description

FIELD OF THE INVENTION

[0001] The invention relates to decoding methods and systems being adapted for execution of these decoding methods for systematic codes such as turbo codes.

BACKGROUND OF THE INVENTION

[0002] Turbo codes belong to the class of *systematic codes* in which the original data stream is transmitted with the encoded stream(s). Turbo encoding is characterised in that the to be encoded signal u_k is encoded by a first encoder ENC1, resulting in a first encoded signal and an interleaved version of u_k is encoded by a second decoder ENC2, resulting in a second encoded signal. The original to be encoded signal u_k, the first and second encoded signal are then transmitted. In the transmitted signal one can thus distinguish a sequence of blocks, being said to be encoded signal u_k, said first and second encoded signal. Figure 4(a) shows such an encoder set-up. It can be said that the complete encoder transforms the original input bits u_k in output symbols c_k, comprising a concatenation of convolutional component codes, being separated by a pseudo random interleaver. It is important to note that in turbo coding reinitialisation of the state of the encoder is essential for the corresponding decoding procedure. Often also part of the input sequence to such an encoder is adapted such that the end state of the encoder is forced to be a particular known state. Note that a pseudo random interleaver has a mapping function having a substantially irregular pattern.

[0003] Turbo decoding is characterised in that the received transmitted code is decoded by a first decoder, resulting in a first decoded signal. After determining said first decoded signal a second decoder does another decoding step, resulting in a second decoded signal, exploiting the received transmitted code and an interleaved version of the first decoded signal. Afterwards said first decoder performs a further decoding step, exploiting said received transmitted cdoe and an interleaved version of said second decoded signal. This procedure is carried out iteratively until the final decoded signal is determined. Said first and second decoded signal can be denoted intermediate data elements, as these are not the final decoded signal. Turbo decoding can be described as performing iterative decoding methods which are embedded in the decoders. The decoding method of a first decoder exploits, in an iteration, data generated by another decoder in the previous iteration. Figure 4(b) shows such a turbo decoding set-up. Said intermediate data elements are denoted also extrinsic information or a posteriori information. The data element determined in a previous iteration is used as intrinsic information in a next iteration. It is assumed as part of the method that performing said iteration results in intermediate data elements which are approximations of a maximum likelihood estimate of the original signal u_k. Because said first and second decoder exploit signals from each other, only one of said decoders is active at a time, meaning that half of the hardware required in a turbo decoding architecture is idle while performing turbo decoding. Known approaches for solving this use pipelining, meaning that a sequence of hardware blocks is used for turbo decoding. A first block performs first decoding, a second block performs second decoding, a third block performs again said first decoding and so on. While said third block performs its first decoding step said first block can alreay start executing its first decoding step on a new sequence. Naturally, such an approach requires a lot of hardware.

[0004] Reference to turbo decoding and decoding methods used therein are found in: [D. Garrett, M. Stan, "Low Power Architecture of the Soft-Output Viterbi Algorithm", Proceedings International Symposium on Low Power Electronics and Design (ISLPED'98), Monterey, CA, August 10-12, 1998, pp. 262-267][O.J. Joeressen, M. Vaupel, H. Meyr, "High-Speed VLSI Architectures for Soft-Output Viterbi Decoding", Journal of VLSI Signal Processing, 1-12, 1998]. [C. Berrou, A. Glavieux, P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," *Proc. ICC '93*, Geneva, Switzerland, May 1993, pp. 1064-1070]. [S.S.Pietrobon, "Efficient Implementation of Continuous MAP Decoders and a Synchronisation Technique for Turbo Decoders", Int. Symp. on Inform. Theory and its Applications, Victoria, BC, Canada, Sep. 1996, pp. 586-589].

[0005] There are several decoding methods used by decoders within a turbo decoding set-up of which one is the Maximum A Posteriori (MAP) approach and will be used in the following as an example but the present invention is not limited thereto. The log-SISO algorithm is chosen as specific algorithm for the description although the invention is not limited hereto, e.g. also Soft-output Viterbi Algorithms can be used. By operating in the logarithmic domain, expensive multiplications are avoided. Instead the E-operation is introduced, which can easily be implemented using table look up or approximated by taking the maximum. The extrinsic information λ_k^{ext} is calculated based on α and β state metrics as indicated in formula 1 in which c^1 and c^2 are the output bits for an encoder state transition from s to s' (see Fig. 17).

$$\lambda_k^{ext} = \underset{s',s \to s'}{E}[\delta_k(s,s')] - \underset{s',s \to s'}{E}[\delta_k(s,s')] \qquad (1)$$

BNSDOCID: <EP___1030457A2_I_>

with $\delta_k(s,s')=\alpha_k(s)+\beta_k(s')+c^1 \cdot \lambda^1{}_k+c^2 \cdot \lambda^2{}_k$
The log likelihood ratios $\lambda_k{}^i$ (for i =1..2) of the channel symbols $y_k{}^i$ are defined as:

$$\lambda_k^i = \log\left[\frac{P(c_k^i=1|y_k^i)}{P(c_k^i=0|y_k^i)}\right]$$

After some iterations the decoded bits $\hat{u}_k$ are calculated as ($\lambda_k{}^{int}$ is the intrinsic information):

$$\hat{u}_k = sign\left[\lambda_k^{ext} + \lambda_k^{int} + \lambda_k^1\right] \quad (2)$$

The α and β metrics are obtained through formula 3 and 4 based on a forward recursion and a backward recursion respectively. They both start in a known initial state at the beginning (for α) or end (for β) of the block.

$$\alpha_{k+1}(s') = \underset{s\to s'}{E}\left[\alpha_k(s)+c^1 \cdot \lambda_k^{int} + c^1 \cdot \lambda_k^1 + c^2 \cdot \lambda_k^2\right] \quad (3)$$

$$\beta_{k-1}(s) = \underset{s\to s'}{E}\left[\beta_k(s')+c^1 \cdot \lambda_k^{int} + c^1 \cdot \lambda_k^1 + c^2 \cdot \lambda_k^2\right] \quad (4)$$

In general, in these MAP algorithms, a compute step and a determining step can be distinguished. Said compute step is characterised by the computation of two vector sequences or state metrics. Said vector sequences are computed via recursions. A forward recursion for determining said first state metrics and a backward recursion for determining said second state metrics are distinguished. Said state metric determination exploits the encoded signal (via $\lambda_k{}^1$, $\lambda_k{}^2$) and intermediate data elements λ^{INT}, produced by another decoder. Said decoded signal u_k is determined by combining said encoded signal (via $\lambda_k{}^1$), said first state metrics and said second state metrics (via λ^{ext}). Note that the coding process can be seen as a transition in a finite state machine, wherein the register content of the convolution coder, denotes the state of the encoder, which completely determines the behaviour of the coder for the next input bit. One often represents this with a trellis graph, showing state transitions. The state metrics exploited in the decoding process refer in principle back to these encoder state.

[0006] The α metrics need to be stored however since the first λ^{ext} can only be calculated once the entire forward recursion is finished. This results is a storage of *N* metrics for all the states, which is unacceptable for most practical interleaver sizes *N*. A solution to the storage requirement problem of the normal SISO algorithm presented above, is the introduction of sliding windows [S.S.Pietrobon, "Efficient Implementation of Continuous MAP Decoders and a Synchronisation Technique for Turbo Decoders", Int. Symp. on Inform. Theory and its Applications, Victoria, BC, Canada, Sep. 1996, pp. 586-589]. The β state metrics are not initialized at the end of the block but at some point *k* (see Fig. 10). After the backward recursion over window size *L* time steps the metrics provide an accurate approximation at time *k-L*. The next metrics β_{k-L} through β_{k-2L} are calculated and used to produce the extrinsic values. The window is then shifted by a value *L*. This algorithm requires the storage of only L α metrics.

[0007] The use of overlapping windows, also denoted sliding windows, comprises the computation of one of said state metrics, with its corresponding recursion being validly initialized, while the other state metric is then determined a plurality of times but each time only part of said state metrics are determined and the recursions used therefore are not validly initialized. Recursions wherein only part of said state metrics and which are not validly initialized are further denoted restricted resursions. The overlapping window approach can then be described as a method wherein one of said state metrics is determined completely with a validly initialized recursion while the other state metric is then determined a plurality of times with restricted recursion, determinining only part of these state metrics. It must be emphasized that although so-called invalid initializations are used, the turbo decoding schemes show the property that after some recursion steps, said computed state metrics converge towards the state metrics expected when valid initializations were exploited. In so-called overlapping window approaches described above either one of said state metrics is determined completely and with a valid initialized recursions.

[0008] The sliding windows approach cures the memory requirement problems only partially. An additional important problem with turbo decoding, either in a standard way or via overlapping windows, is the long latency, due to the intrinsic iterations in combination with the long recursions.

BNSDOCID: <EP___1030457A2_I_>

AIM OF THE INVENTION

[0009] It is an aim of the invention to provide methods for decoding of systmeatic codes such as turbo codes which are low power consuming, which have reduced memory requirements and showing enhanced performance with respect to the latency problem.

[0010] It is a further aim of the invention to provide architectures being adapted for execution of such low power consuming, less memory requiring methods with lower latency.

SUMMARY OF THE INVENTION

[0011] In the invention aspects related to the overall turbo decoding approach and aspects related to the particular decoding approach used within such a turbo decoding approach can be distinguished. It must be emphasized that, although the particular decoding approach aspects of the invention are situated within the overall turbo decoding approach in the description, the contribution of each of these aspects must be recognized in general.

[0012] Turbo decoding schemes can be characterised as methods for determining a decoded signal from an encoded signal which has been encoded by a turbo encoding scheme. In such a turbo decoding scheme, besides a step of inputting or entering said encoded signal, a compute step and a determining step can be distinguished. Said determining step can be performed simultaneously, partly overlapping or after said with said compute step. In turbo decoding schemes said compute step is characterised by the computation of two vector sequences. One vector of such a sequence is denoted a state metric. Therefore in said compute step state metrics are determined. With first state metrics is meant a first vector sequence. With second state metrics is meant a second vector sequence. Said vector sequences are computed via recursions. With recursion is meant that the following vector in such a vector sequence is determined by at least the previous vector in said sequence. In turbo decoding schemes a forward recursion for determining said first state metrics and a backward recursion for determining said second state metrics are distinguished. The terms forward and backward refer to the order in which said encoded signal is inputted. Turbo decoding schemes are characterised by the fact that said decoded signal is determined by combining said encoded signal, said first state metrics and said second state metrics.

[0013] In a first aspect of the invention particular ways of storing said vector sequences or state metrics in memories are presented. Indeed when one wants to determine said decoded signal from said encoded signal and said state metrics, and when said state metrics are not produced or computed at the same time as these state metrics are needed for consumption, storage of already computed state metrics is a possible way to go. As said vector sequences, exploited in turbo decoding schemes, are typically long, large memories, being power consumptive and long access times, are then needed. As low power implementation and low latency of turbo decoding schemes is aimed at in the invention, an alternative approach is presented. After inputting said encoded signal, said first state metrics is determined by a forward recursion. Said forward recursion is properly initialised. Said forward recursion exploits said inputted encoded signal. However not all said computed first state metrics or vectors are stored in a memory. Note that all said first state metrics must be computed one after another due to the forward recursion approach. In the invented approach however only a part of said necessarily computed first state metrics is stored in a memory, denoted a first memory. With storing part of said computed first state metrics is meant that the amount of stored values is less then the total size or length of said vector sequence or first state metrics which have been generated. In practice it is meant that an amount of stored values is substantially less than the total length. After computing said first state metrics and storing part of them, said second state metrics are computed with a backward recursion. When a particular state metric of said backward determined state metrics becomes available, it can be almost directly exploited for determining said decoded signal from said encoded signal, said second state metrics and said computed first state metrics. Said second state metrics thus do not need a large buffer memory as data consumption is scheduled near in time to data generation. Said invented approach can be characterised in that only a part of said computed first state metrics is stored in a first memory, more in particular, in a memory being substantially smaller than the size or length of said first state metric sequence.

[0014] In an embodiment of this first aspect of the invention said decoded signal is determined by exploiting a calculation step, wherein said decoded signal is determined directly form said encoded signal, said second state metric and said computed first state metrics, being stored in said first memory.

[0015] In another embodiment of this first aspect of the invention said decoding signal is determined from said encoded signal, said second state metrics and the needed first state metrics, itself being determined or recomputed. With said recomputation is preferably not meant that said first state metrics are determined all over again, starting from the initialisation. With recomputation is preferably meant that first state metrics, having already been computed but not stored, are recomputed from first state metrics which have been computed and stored. More in particular, first state metrics, not stored and lying in between or intermediate stored first state metrics are determined from the first state metrics, bounding the sequence of not-stored ones. One can then state that in said decoded signal determining step recomputed first state metrics are used.

BNSDOCID: <EP___1030457A2_I_>

[0016] In a further embodiment of this first aspect of the invention wherein said non-stored first state metrics are recomputed one does not necessarily consume said recomputed values directly when producing or recomputing them. Further, it is not necessary that said non-stored first state metrics are recomputed several times when needed. Indeed such unnecessary recomputation or direct production-consumption restriction can be circumvented by at least partly storing said recomputed values in a second memory. In practise said recomputed values will be stored only temporarily in said second memory. The size of said second memory will be substantially less than the size of said first state metric sequence. In an embodiment of the invention said second memory size will be equal or even smaller than would be required to store the amount of non-stored first state metrics in between said stored ones. Said second memory can thus contain at most said intermediate first state metrics. This size constraint on said second memory can result in overwriting previously stored recomputed values which are no longer required.

[0017] Alternatively, one can state that in the invention instead of storing the β metrics, being determined with a backward recursion, for all time steps k, only some are stored and the missing ones are recalculated when they are needed to compute λ^{out}. When only $1/\theta$ of the backward state metrics, are stored this means that only $\{\beta_i(S_0), \beta_i(S_1), \ldots, \beta_i(S_n)\}$, $\{B_{i+\theta}(S_0), \beta_{i+\theta}(S_1), \ldots, \beta_{i+\theta}(S_n)\}$, $\{\beta_{i+2\theta}(S_0), \beta_{i+2\theta}(S_1), \ldots, \beta_{i+2\theta}(S_n)\}$, ... are stored in memory. The parameter θ is determined by simulations, taking into account the architecture on which the algorithm should be implemented, the appropriate power and area models and a cost criterion which may be determined by area, energy consumption and latency. Note that alternatively the same approach can be used for the state metric being determined with a forward recursion.

[0018] In a second aspect of the invention, particular ways of executing said state metric recursions and said decoded signal determination step are presented aimed at providing methods wherein a trade-off between latency and memory occupation can be made. In a traditional execution of a turbo decoding scheme one computes via a forward recusion said first state metrics, being validly or correctly initialized, and then one computes said second state metrics, being validly initialized, with a backward recursion. Simultaneously with, or after said second state metric computation, one determines said decoded signal. Executing turbo decoding schemes in such a way results in a long latency and large memories for storage of said state metrics and said inputted encoded signal. Another approach, denoted also as the use of overlapping windows, comprises the computation of one of said state metrics, with its corresponding recursion being validly initialized, while the other state metric is then determined a plurality of times but each time only part of said state metrics is determined and the recursions used therefore are not validly initialized. Recursions wherein only a part of said state metrics is determined and which are not validly initialized are further denoted restricted recursions. The overlapping window approach can then be described as a method wherein one of said state metrics is determined completely with a validly initialized recursion while the other state metric is then determined a plurality of times with restricted recursion, determinining only part of these state metrics. It must be emphasized that although so-called invalid initializations are used, the turbo decoding schemes show the property that after some recursion steps, said computed state metrics converge towards the state metrics expected when valid initializations were exploited. In so-called overlapping window approaches described above, either one of said state metrics is determined completely and with a valid initialized recursions. Such approaches still show a long latency. In the invention a method for turbo decoding is presented wherein for both state metrics (first and second) only part of these state metrics is determined with a validly initialized recursion, while the other parts, thus the one not determined by a validly initialized recusion, are being determined by restricted recursions which preferably approach valid values asymptotically but quickly. More in particular a plurality of said restricted recursions are needed. Execution of part of said restricted recursions are performed at the same time. The invented approach is further denoted a double flow approach.

[0019] In an embodiment of this second aspect of the invention, the decoded signal determining step is performed while executing said validly initialized recursions. An example is given now, but it must be clear that the role of said first and said second state metric can be reversed. In such an approach, one starts a first restricted recursion for determining with a backward recursion part of said second state metric. After some recursion steps, valid values of said second state metric are obtained. When the second state metric to be determined by said restricted recursion are all determined, one starts computing said first state metric, with said validly initialized forward recursion and one consumes substantially simultaneously the computed first state metric, the already determined second state metric, if valid, and the encoded signal, in order to determine the decoded signal. After a while no valid second state metrics from said first restricted recursion are available. Therefore a second restricted recursion, preferably started in advance of its need, now provides said second valid state metrics. The first state metric is still provided by said validly initialized first state metric. The same approach is used for said second state metric. A validly initialized backward recursion is started, supported with restricted recursion of said first state metrics. It must be emphasized that said validly initialized recursions, with their supporting restricted recursions, are essentially dealing with other parts of said state metrics. Said validly initialized recursions stop when they reach same point in the vector sequence of state metrics, indicating that the full range of state metrics is covered.

[0020] In another embodiment of the invention of the second aspect of the invention said decoding signal determining step is performed while executing said non-validly initialized recursions. An example is given now, but it must be

BNSDOCID: <EP___1030457A2_I_>

clear that the role of said first and said second state metric can be reversed. In such an approach one starts a first restricted recursion for determining with a backward recursion part of said second state metric. Simultaneously, one starts computing said first state metric, with said validly initialized forward recursion but the computed first state metric values are not directly consumed for determining said decoded signal. After some recursion steps valid values of said second state metric are obtained. Then one further determines with said restricted recursion second state metrics and one consumes substantially simultaneously the already computed first state metric, the further determined second state metric and the encoded signal, in order to determine the decoded signal. The first state metrics are also further determined with said forward recursion. After a while the part of the decoded signal that can be determined from said second and first state metrics is found. Therefore, a second restricted recursion, being started already, now provides said second valid state metrics, while the first state metrics were already determined by said continuing forward recursion. The same approach is used for said second state metric. A validly backward recursion is started, supported with restricted recursion of said first state metrics. It must be emphasized that said validly initialized recursions, with their supporting restricted recursions, are essentially dealing with other parts of said state metrics. Said validly initialized recursions stop in this case not when they reach the same point in the vector sequence but soon thereafter. Indeed some part of the decoded signal still has to be determined. In this last phase the validly initialized recursion deliver the state metrics to be consumed for decoded signal determination.

[0021] In another embodiment of the invention of the second aspect of the invention said decoded signal determining step is being performed partly while executing part of said non-validly initialized recursion and partly while executing said validly initialized recursion. This embodiment can be seen as a mixed form of the two embodiment described above. Again two validly initialized recursions, one for each state metric, are executed. Each of said validly initialized recursions is supported by so-called restricted recursions of the complementary state metric. Determining a value of the decoded signal is possible when both corresponding state metrics are available and of course valid. In this embodiment the first restricted recursions are started a substantial time before said validly initialized recursions begun.

[0022] In a third aspect of the invention particular methods for turbo decoding, comprising essentially of turbo decoding method steps of smaller sizes are presented. A turbo decoding method determines in principle a decoded signal by combining an encoded signal, first state metrics and second state metrics. Said metrics are computed via a forward or a backward recursion. Said recursions in principle need to be validly initialized. However because even invalidly initialized recursions converge after some dummy state metric computations towards valid values of state metrics, one can work with invalidly initialized recursions also, as pointed out in the embodiments discussed above. When for both recursions invalid initializations are used, one can split up the turbo decoding algorithm into a plurality of turbo decoding method steps of smaller sizes, meaning that the vector sequences or state metrics in such methods are of smaller length. In principle said state metrics of such method of smaller size together define the fill state metrics, except that also some dummy metrics are computed. It can be stated that the turbo decoding method comprises of executing a plurality of compute and determining steps. Each of said compute and determining steps comprising computing part of said first state metrics with a forward recursion and part of said second state metrics with a backward recursion, and determining part of said decoded signal by combining part of said encoded signal, part of said first state metric and part of said second state metric. Although said compute and determining steps can be seen as totally separate methods, in the invention particular scheduling of these compute and determining steps are proposed such methods support each other, by providing valid initializations to each other.

[0023] In an embodiment of this third aspect of the invention at least two of these compute and determining steps are executed or scheduled such that these are performed partially overlapping in time.

[0024] In another embodiment of this third aspect of the invention at least two of said compute and determining steps are scheduled such that initialisation of one of said recursions for said computing state metrics of one of said compute and determining steps is based on a computed value of said recursion of said state metrics in the other compute and determining step.

[0025] In the fourth aspect of the invention a method for iterative decoding is presented. Note that iterative decoding is typically used in turbo decoding. In principle the presented iterative decoding method can be exploited in a context different to turbo decoding also. Turbo decoding is characterized by performing a first decoding step followed by a second decoding step, and iteratively performing these two decoding steps. After loading or inputting the encoded signal, a first decoding step is performed, which produces or determines a first intermediate data element. The second decoding step exploits an interleaved version of said first intermediate data element in order to produce or determine a second intermediate data element. With interleaved version is meant that essential the same values are stored in said data element but the ordering within said data element, being a vector, is changed. After this both said first and said second decoding step are performed iteratively but never simultaneously until from one of said data elements said encoded signal can be deduced. Note that this procedure implies that the hardware performing said first decoding step and the hardware performing said second decoding step are not active simultaneously, thus an inefficient use of the overall hardware is obtained. Note that a standard turbo decoding set-up, as depicted in Figure 4(b), requires two memories as interleavers. In the invention an iterative decoding procedure, which can be used in turbo decoding but is not limited

BNSDOCID: <EP___1030457A2_I_>

thereto, enabling more efficient use of hardware, is presented. The invented method is based on performing said first and second decoding step on the same hardware, said first and second decoding step exploiting the same single-port memory. Said hardware and said single-port memory are used in a feedback configuration. Note that the determining of the encoded signal by the decoding methods exploited in said iterative procedure can be denoted in this context soft decisions as in fact only an intermediate data element is determined. It is in the final step of the iterative procedeure, when actually determining the best estimate of the decoded signal, that a hard final decision is taken.

[0026] In an embodiment of this aspect of the invention it can be recognized that by considering said iterative performing said first decoding step and second decoding step together as a method step and applying said method step only on part of said encoded signal, then this method step will be smaller in size, thus needing smaller memories. The turbo decoding procedure can then be done by performing a plurality of such smaller size method steps, each comprising of iteratively performing first and second decoding steps, substantially simultaneously. Each of such smaller size method steps then has a memory, being assigned to it, and used for storage of the intermediate data elements, produced by it. Said memories are of the single-port type.

[0027] In a further embodiment of this aspect, the use of a flexible address generator is described. Indeed as said single-port type memory or storage unit must show interleaver functionality, flexible generation of addresses for storage in and retrieval out said memory is needed.

[0028] In a further embodiment of this aspect of the invention further iterative performing of a plurality of decoding steps instead of said first decoding step and second decoding step only is described.

[0029] In another embodiment of this fourth aspect of the invention one recognizes said iterative performing said first decoding step and second decoding step are considered together as a method step and by executing a plurality of such methods substantially simultaneously, one can decoded more than one sequence of blocks, related to said encoded signal, at a time, which again increases the efficiency of the hardware used. Naturally, such an approach is not limited to two decoding steps but can be extended to a plurality of decoding steps.

[0030] In a fifth aspect of the invention essentially digital devices, being capable of executing any one of the turbo decoding methods described above, are presented, a digital device capable of executing one of the above methods being an embodiment of the present invention. Said digital devices are adapted such that they can either use said partial storage and recomputation based method and/or implement said so-called double flow methods and/or exploit parallelism and/or comprise of the hardware-memory feedback configurations discussed before.

[0031] In a sixth aspect of the invention parametric optimization of turbo decoding methods, being particularly adapted for providing sufficient degrees of freedom, is presented.

BRIEF DESCRIPTION OF THE DRAWINGS

[0032]

Fig. 1 shows a general legend which will be used for the representation of the turbo decoding methods in the several aspects and embodiments of the invention. The horizontal dimension is related to execution time, while the vertical dimension relates to the index number of the state metric involved. Moreover this kind of representation provides in a clear way the particular ways in which the turbo decoding methods presented, deal with dummy or invalid metrics, valid and stored but not consumed metrics and valid and substantially direct consumed state metrics. This representation also allows representing in a visual way the forward recursion and the backward recursion involved in these turbo decoding methods.

Figs. 2(a) to (c) each show an individual embodiment of the so-called double flow turbo decoding method, which is characterised in that both said state metrics are only partially calculated with a validly initialized recursion which is indicated with a solid line.

Figs. 3(a) and (b) illustrate recalculation methods exploited in the presented turbo decoding methods. Figs.3(a) and (b) show a method wherein a metric is partially stored in a metrics memory while it is computed during its generating recursion and while another metric is generated and consumed the not-stored values which are located in between said stored metric are recomputed. In Fig. 3(a) said recomputed metric is stored while in Fig. 3(b) recomputation is each time restarted from a stored metric. The figure illustrates that in an apparatus being adapted for executing said methods, memory for storage of partially stored metrics and possibly memory for storage of recomputed metrics are provided. Said methods are characterised in that they do not store the full state metric sequence.

Fig. 4 shows a prior art turbo encoder (a) and a turbo decoder (b). One observes in said decoder two decoder modules (DEC1, DEC2) which are exploiting values from each other via the interleaver (π and π^{-1}). Said decoder modules each perform substantially different decoding steps, each decoder module being assigned a particular decoding step.

Fig. 5 (a) shows the operation of a turbo decoder as shown in Fig. 4(b). A first decoder consumes values according to a first decoding method and stores these in a first order in a memory. The second decoder consumes in a further

BNSDOCID: <EP___1030457A2_I_>

step said produced values in a second order, being substantially different from said first order, and produces values according to a second decoding method, substantially different from said first decoding method and stores them in a memory. Fig. 5(b) illustrates the operation of a turbo decoder according to another embodiment of the present invention. A turbo decoder setup is provided wherein a single decoder module, performing a decoding method, has a single memory, being organized in a feedback arrangement. The right and left picture in Fig. 5(b) show the operation of the same decoder module but during a different time slot. Note that said decoder module can thus execute two substantially different decoding steps.

Fig. 6 shows a turbo decoder setup wherein the principle of Fig. 5(b) is expanded by showing a plurality of decoding modules, each being arranged in a feedback arrangement with a memory dedicated or assigned to each decoding module. Further multiplexers and demultiplexers are provided. All of said decoding modules are adapted such that they can execute a plurality of decoding methods in different time slots. The proposed set-up or apparatus is an essentially digital device and is particularly suited for executing the turbo decoding methods proposed in the different aspects and embodiments of the invention.

Fig. 7 shows another representation of a turbo decoder setup in accordance with an embodiment of the present invention.

Fig. 8 illustrates a prior-art execution of a typical decoding method, being exploited in turbo decoding. A first forward recursion and a second backward recursion are provided, both being validly initialized and covering the full length N of the state metrics. Said first forward recursion is characterized in that the state metrics being determined are not consumed for determining said decoded signal immediately, as indicated by the dashed line. The shading of the figure indicates storage requirements.

Fig. 9 illustrates a compute and determining step being exploited in turbo decoding methods according to the invention. Said compute and determining step shows also a first forward and a second backward recursion, but both said recursions are not validly initialized and are not covering the full length N of the state metrics but only a part S substantially smaller then N. Note that L refers to the dummy metrics sequence, indicated by dotted lines, needed due to invalid initializations.

Fig. 10 (a) shows again the prior art execution of a typical decoding method as explained in Fig. 8. Fig. 10(b) illustrates an embodiment of a turbo decoding method according to the invention, wherein a plurality of said compute and determining steps are executed substantially simultaneously.

Fig. 11(a) shows two substantially simultaneously executed compute and determining steps, being reversed with respect to each other, and being scheduled such that they provide valid initializations to each other. The light dotted line indicates dummy state metrics, which do not need to be determined anymore due to this particular arrangement. The memory requirements of this approach are indicated by the shading. Fig. 11(b) shows two time deferred and thus partially overlapping in time compute and determining steps, being reversed with respect to each other, and being scheduled also such that they provide valid initializations to each other. This arrangement is such that said two compute and determining steps can share a common storage unit, because their peak in memory requirement as function of time is at a substantially different time. Again memory occupancy is indicated by shadings.

Fig. 12 shows another embodiment of the invention wherein several compute and determinings step are executed in a particular arrangement with respect to each other in order to facilitate transfer of valid initializiations on the one hand and improving the memory requirements. Here 6 compute and determining steps are shown. This embodiment can also be parametrized as having (M=) 6 windows and (K=) 3 sequential valid state metric sequence calculations after each other.

Figs. 13(a) and (b) show embodiments of the invention with parameter values M=2, K=3. The memory size per dummy sequence is here 0.5 and 1 dummy sequence per output sequence.

Figs. 14 (a) to (c) show embodiments of the invention with parameter value K=4. The memory size per dummy sequence and the amount of dummy sequence per output sequence is indicated.

Fig. 15 shows embodiments of the invention with parameter value K=5. The memory size per dummy sequence and the amount of dummy sequence per output sequence is indicated.

Fig. 16 shows the first state metrics ALPHA, being determined with a forward recursion, are being divided in sets or classes (1)(2)(3). The second state metrics BETA, being determined with a backward recursion, are also divided in sets or classes (1)(2)(3). The first state metrics of one of such sets and the second state metrics of the corresponding set are computed both by a compute and determining step (4)(5). There are a plurality of such compute and determining steps. Observe that the compute and determining step (4) provides an input (6) to compute and determining step (5), more in particular for providing a valid initialization to step (5). Note that said compute and determining steps (4)(5) are being scheduled such that said steps are executed at least partially overlapping in time.

Fig. 17 illustrates an interpretation of the two state metrics, in a so-called trellis graph.

Fig. 18 is an alternative representation as in Fig. 1. Here the valid initializations are indicated with electrical grounding symbols.

BNSDOCID: <EP___1030457A2_I_>

Fig. 19 presents the so-called sliding window approach.

Figs. 20(a) to (e) show alternative representations of decoding methods in accordance with the present invention. In particular Fig. 20(c) represents a double flow decoding method in accordance with the present invention.

Fig. 21 shows another representation of the so-called double flow approach. Two classes of state metrics are distinguished. The first state metrics, to be determined with a forward recursion, is denoted ALPHA while the second state metrics, to be determined with a backward recursion, is denoted BETA. Both classes can be further separated in subclasses. One subclass contains metrics (4) which are determined by a validly initialized recursion (1)(2) while the other subclass contains metrics (5) which are not determined by such a validly initialized recursion. These metrics (5) are then determined by a plurality of recursions (3), also denoted restricted recursions, which are not validly initialized and which themselve only determine part of the metrics (5).

DETAILED DESCRIPTION OF THE INVENTION

[0033] The present invention will be described with reference to certain embodiments and to certain drawings but the invention is not limited thereto but only by the claims.

[0034] Turbo decoding is characterised in that the received transmitted code is decoded by a first decoding step, resulting in a first decoded signal. Turbo decoding does not stop however after determining said first decoded signal. Instead a second decoding step resulting in a second decoded signal, is performed, said second decoding step exploits the received transmitted code and an interleaved version of the first decoded signal. Afterwards again said first decoding step is performed, exploiting said received transmitted code and an interleaved version of said second decoded signal. This procedure is iteratively and until the final decoded signal is determined.

[0035] Thus, within turbo decoding schemes decoding steps are used. These decoding steps can be characterised as methods for determining a decoded signal from an encoded signal, being encoded by a turbo encoding scheme. In such a turbo decoding scheme, besides a step of inputting said encoded signal, a compute step and a determining step can be distinguished. Said determining step can be carried out simultaneously, partly overlapping or after said compute step. In turbo decoding schemes said compute step is characterised by the computation of two vector sequences. One vector of such a sequence is denoted a state metric. Therefore in said compute step state metrics are determined. With first state metrics is meant a first vector sequence. With second state metrics is meant a second vector sequence. Said vector sequences are computed via recursions. With recursion is meant that the following vector in such a vector sequence is determined from at least the previous vector in said sequence. In turbo decoding schemes a forward recursion for determining said first state metrics and a backward recursion for determining said second state metrics are used. The terms forward and backward refer to the order in which said encoded signal is inputted. Turbo decoding schemes are characterised by the fact that said decoded signal is being determined by combining said encoded signal, said first state metrics and said second state metrics.

[0036] In the further description said first state metric is denoted α, while said second state metric is denoted β. α and β are vectors with a fixed dimension and each part of a vector sequence. Thus a vector sequence of α's and a vector sequence of β's can be distinguished. To each vector in such a sequence a number is assigned, indicating the place of said vector in its corresponding sequence.

[0037] Aspects and the embodiments of the invention can be illustrated practically with graphical representations as shown in Fig. 1 and which will be used in the following figures except Fig. 18. The horizontal dimension of these representations relate to time. The more to the right a point is in each of these representations, the later it occurs. The vertical dimension of these representations relate to the number of each vector in its correspondig sequence. A lower point in each of these representations has a lower number than the higher points in such a representation. The representations in Fig. 1 can thus be used for representing how the state metrics α and β are computed and exploited as function of time. Note that α in Fig. 1 relates to the state metric being determined with a forward recursion while β is related to the state metric being determined with a backward recursion. Interchanging the notation results in equivalent representations. In the above description a difference is made between (a) computing state metric, not storing and also not exploiting them, because they are invalid, (b) computing state metrics and storing them but not exploiting them for determining the decoded signal, and (c) exploiting them for determining said decoded signal. In Fig. 1 a) to c) are indicated by dotted, dashed and full lines, respectively. Fig. 18 is an alternative representation, wherein electrical grounding and an opposite electrical grounding symbol, denote valid or correct initializations.

[0038] In a first aspect of the invention particular ways of storing said vector sequences or state metrics in memories are presented. When one wants to determine said decoded signal from said encoded signal and said state metrics, and when said state metrics are not produced or computed at the same time as these state metrics are needed for consumption, storage of already computed state metrics is a possible way to go. As said vector sequences, exploited in turbo decoding schemes, are typically long, large memories, being power consumptive, are then needed. As low power implementation of turbo decoding schemes is an aim of the present invention, an alternative approach is presented. Alter inputting said encoded signal, said first state metrics is determined by a forward recursion. Said forward recursion

BNSDOCID: <EP___1030457A2_I_>

is properly initialised. Said forward recursion exploits said inputted encoded signal. However, not all said computed first state metrics or vectors are stored in a memory. Note that all said first state metrics must be computed one after another due to the forward recursion approach. In the invented approach however only part of said necessarily computed first state metrics is stored in a memory, denoted a first memory. With storing part of said computed first state metrics is meant that the amount of stored values is less then the total size or length of said vector sequence or first state metrics. In practice this means that an amount of stored values being substantially less then the total length is stored. After computing said first state metrics and having stored part of them said second state metrics are computed with a backward recursion. When a particular state metrics of said backward determined state metrics becomes available, it can be almost directly exploited for determining said decoded signal from said encoded signal, said second state metrics and said computed first state metrics. Said second state metrics thus do not need a large memory as their consumption is scheduled near in time to its production. Said invented approach can be characterised in that only a part of said computed first state metrics is stored in a first memory, more in particular, in a memory being substantially smaller than the size or length of said first state metric sequence.

[0039] Fig. 3 gives graphical representations of the presented approaches. The top array shows the state metrics β or the vector sequences of β's. The higher the ordering number of a β the more the β is located at the right. The vector sequence is determined by initializing the most right β and by determining β's with a lower number from β's with a higher number, with a backward recursion. Fig. 3 illustrates with the shaded boxes that only a part of the computed β's are stored in memory. It must be emphasized that the indicated top array is thus not completely stored in memory. When the vector sequence of β's is determined, the computing of the α sequence begins by initialising the a metric with the lowest number and using a forward recursion. If one wants to determine the decoded signal from the inputted encoded signal and both state metrics, one also needs the corresponding β, meaning the β with the corresponding number. As said β's are not all stored, some alternative approach must be used.

[0040] A possible approach in accordance with the present invention is to determine the decoded signal directly from the stored β's. This requires another type of computation or calculation. In fact one needs to compute a mathematical expression relating the decoded signal directly to said stored β's. The extra computational effort, due to the increase in complexity of the mathematical expression to be computed, must be compared with the cost reduction due to the reduced memory requirements. It must be emphasized that in the representation of Fig 3 α and β can be reversed. Therefore, this embodiment of this first aspect of the invention can be described also by stating that said decoded signal is determined by exploiting a calculation step, wherein said decoded signal is determined directly form said encoded signal, said second state metric and said computed first state metrics, being stored in said first memory.

[0041] Another possible approach in accordance with the present invention is to determine the decoded signal from β's, which are recalculated or recomputed. Representation (b) in Fig. 3 shows an approach wherein one determines the decoded signal from left to right, using forward recursions for α and one computes each time the β's when needed. One notes that as one has to start from a stored one and because these β's are computed with backward recursions, several compute steps are needed. Note that a small register file is exploited here for storage of one temporary β value. It must be emphasized that in the representation of Figure 3 α and β can be reversed. Therefore this embodiment of this first aspect of the invention can be described also by stating that said decoding signal is determined from said encoded signal, said second state metrics and the needed first state metrics, these being determined or recomputed. With said recomputation is not meant that said first state metrics are determined all over again, starting from the initialisation. With recomputation is meant that first state metrics, being computed before but not stored, are recomputed from first state metrics, which have been computed and stored. More in particular first state metrics, not stored and lying in between or intermediate stored first state metrics are determined from the first state metrics, bounding the sequence of not-stored ones. One can then state that in said decoded signal determining step, explicitly recomputed first state metrics are used.

[0042] Another possible approach to determined the decoded signal from β's, which are recalculated or recomputed. Representation (a) in Fig. 3 shows an approach wherein one determines the decoded signal from left to right, using forward recursions for α and one computes the β's when needed but also temporary stores them. One notes that as one has to start from a stored one and because these β's are computed with backward recursions, several compute steps are needed. Because now temporary storage is used instead of a small register for storage of one temporary β value only, the amount of compute steps is reduced when compared to the approach of representation (b). It must be emphasized that in the representation of Figure 3 α and β can be reversed. Therefore this embodiment of this first aspect of the invention can be characterized by stating said non-stored first state metrics are recomputed but one does not necessarily consume said recomputed values directly when producing or recomputing them. One stores said recomputed values in a second memory. In practice said recomputed values will be stored only temporarily in said second memory. The size of said second memory will be substantially less than the size of said first state metric sequence. In an embodiment of the invention said second memory size will be equal or even less than the amount of non-stored first state metrics in between said stored ones. Said second memory can thus contain at most said intermediate first state metrics. The size constraint on said second memory results in overwriting said stored recomputed values.

BNSDOCID: <EP___1030457A2_I_>

[0043] In a second aspect of the invention particular ways of execution said state metric recursions and said decoded signal determination step are presented aimed at providing methods wherein a trade-off between latency and memory occupation can be made. In a traditional execution of a turbo decoding scheme one computes via a forward recusion said first state metrics, being validly initialized, and then one computes said second state metrics, being validly initialized, with a backward recursion. Simultaneously or after said second state metric computation, one determines said decoded signal. Executing turbo decoding schemes in such a way results in long latency and huge memories for storage of said state metrics and said inputted encoded signal. Another approach, denoted also as the use of overlapping windows, comprises the computation of one of said state metrics, with its corresponding recursion being validly initialized, while the other state metric is then determined a plurality of times but each time only part of said state metrics is determined and the recursions used therefore are not validly initialized. Recursions wherein only a part of said state metrics is determined and which are not validly initialized are further denoted restricted resursions. The overlapping window approach can then be described as a method wherein one of said state metrics is determined completely with a validly initialized recursion while the other state metric is then determined a plurality of times with restricted recursion, determinining only part of these state metrics. It must be emphasized that although so-called invalid initializations are used, the turbo decoding schemes show the property that after some recursion steps, said computed state metrics converge towards the state metrics expected when valid initializations were exploited. In so-called overlapping window approaches described above either one of said state metrics is determined completely and with a valid initialized recursions.

[0044] In the invention a method for turbo decoding is presented wherein for both state metrics (first and second) only part of these state metrics is determined with a validly initialized recursion, while the other parts, thus the one not determined by a validly initialized recusion, are being determined by restricted recursions. More in particular a plurality of said restricted recursions are needed. Execution of part of said restricted recusions are performed at the same time. Fig. 2 illustrates embodiment of this second aspect of the invention graphically. Note that dotted lines represent state metric computations which are not stored as they are invalidly initialized. The shown embodiment is characterized by the fact that for both said state metrics at least one recursion is validly initialized. In the representations in Fig. 2 this is done by starting the forward recursion at the beginning of the corresponding vector sequence, thus at the bottom of the representation and the backward recursion at the end of the corresponding vector sequence, thus at the top of the representation. One recognizes these validly initialized recursions (starting at the top or at the bottom) do not cover the whole vector sequence, thus only part of the state metrics are determined by such a recursion. The other parts must then be determined by other recursions, here denoted restricted recursions, as they only determine part of said state metrics and are invalidly initialized. One recognizes these restricted recursions as combined dotted-dashed lines. Note that said validly initialized recursion can either only compute and store or also compute and determine the decoded signal, depending on the particular embodiment. Note that each embodiment requires another size of memories for storing state metrics, being computed but not consumed yet and for temporary storage of the inputted encoded signal, which is also needed for determining state metrics and for determining said decoded signal. Fig. 2 gives an indication of the size of the required memories. Fig. 20 (c) shows an alternative representation of the embodiment above, by only showing the recursion of the validly initialized metric.

[0045] Fig. 21 shows another representation of this second aspect of the invention. Two classes of state metrics are distinguished. The first state metrics, to be determined with a forward recursion, is denoted ALPHA while the second state metrics, to be determined with a backward recursion, is denoted BETA. Both classes can be further separated in subclasses. One subclass contains metrics (4) which are determined by a validly initialized recursion (1)(2) while the other subclass contains metrics (5) which are not determined by such a validly initialized recursion. These metrics (5) are then determined by a plurality of recursions (3), also denoted restricted recursions, which are not validly initialized and which themselve only determine part of the metrics (5). Although not explicitly drawn in Fig. 21 it must be emphasized that these recursions can partly determine the same metrics (5), although at another time instance. Thus these restricted recursion can overlap with respect to the range of the state metrics. The righthand side of Fig. 21 shows the scheduling of the different recursions. It is shown that said validly initialized recursions (1) and (2) are executed substantially simultaneously. It is also shown that said restricted recursions (5) are partly executed simultaneously. Note that said restricted recursions always determine in a sequence of first iterations dummy or invalid state metrics and a further sequence then valid state metrics. More in particular one relies here on the property that such an non-validly initialized recursion asymptotically converges towards valid state metrics and when the difference between state metrics determined by such a restricted recursion and the theoretically expected ones is below a certain threshold said restriced recursion state metrics are considered to be valid.

[0046] In a third aspect of the invention particular methods for turbo decoding, comprising essential of turbo decoding method steps of smaller sizes are presented. A turbo decoding method determines in principle a decoded signal by combining an encoded signal, first state metrics and second state metrics. Said metrics are computed via a forward or a backward recursion. Said recursions in principle need to be validly initialized. Fig. 8 shows a graphical represention of the above described turbo decoding scheme, wherein the first state metric α, being a part of a vector sequence of

BNSDOCID: <EP___1030457A2_I_>

length N, is computed with a forward recursion but not consumed for determining the decoded signal. In the drawing this is represented by the dashed lines. When the first state metric is determined one starts computing β with a backward recursion and consumes it for determining the decoded signal. In the drawing the full line is used for this representation. One observers a latency along the horizontal axis, indicating time, propertional to 2N. The shaded area indicates memory requirements as function of time needed for storage of the first state metrics.

[0047] However because even invalidly initialized recursions converge after some dummy state metric computations towards valid values of state metrics, one can work with invalidly initialized recursions also, as pointed out in the embodiments discussed above. Fig. 9 shows a representation thereof. A first state metric is determined with a forward recursion, starting with an invalid initialization. One computes some dummy metrics, indicated by the dotted lines, and obtains after a while valid metrics, not consumed yet, as indicated by the dashed lines. The second state metric is determined with a backward recursion, starting with an invalid initialization. One computes some dummy metrics, indicated by doted lines, and obtains after a while valid metrics, which can be consumed directly, indicated by the full line, as the corresponding first state metric is also already available. Note that a valid interval of length S and dummy sequences of length L can be distinguished.

[0048] When for both recursions invalid initializations are used, one can split up the turbo decoding method into a plurality of turbo decoding method steps of smaller sizes, meaning that the vector sequences or state metrics in such methods are of smaller length. In principle said state metrics of such method of smaller size together define the full state metrics, except that some dummy metrics are computed also. It can be stated that the turbo decoding method comprises of executing a plurality of compute and determining steps. Each of said compute and determining steps comprising of computing part of said first state metrics with a forward recursion and part of said second state metrics with a backward recursion, and determining part of said decoded signal by combining part of said encoded signal, part of said first state metric and part of said second state metric. Fig. 10 (b) shows such parallel execution in comparison with the standard method, shown in Fig. 10 (a). Note that the latency is reduced here and is propertional with L+2S.

[0049] Although said compute and determining steps can be seen as totally separate methods, in the invention particular scheduling of these compute and determining steps are proposed such that said methods support each other, by providing valid initializations to each other. Fig. 11 gives graphical representations of these.

[0050] In an embodiment of this third aspect of the invention at least two of these compute and determining steps are executed or scheduled such that these are performed partially overlapping in time. Fig. 11 (b) shows this embodiment. Fig. 11 (a) is an extreme case of this, as both compute and determining steps are executed simultaneously there.

[0051] In another embodiment of this third aspect of the invention at least two of said compute and determining steps are scheduled such that initialisation of one of said recursions for said computing state metrics of one of said compute and determining steps is based on a computed value of said recursion of said state metrics in the other compute and determining step. Fig. 11 (b) shows that the first compute and determining step, shown by the bottom left triangle, produces valid α values, with can be used for initialization of the top right triangle, being a second compute and determining step. Therefore said second compute and determining step does not need to compute the dummy values, indicated by the light dotted lines.

[0052] Note that in Fig. 11 the shaded area indicates the memory requirements as function of time. One observes the trade-off between gaining some calculation effort (dummy metric computation), latency and memory requirements.

[0053] In a further embodiment of this third aspect of the invention one recognizes that intelligent scheduling of compute and determining steps is not necessarily limited to two of said steps. Scheduling of these steps such that these are at least partially overlapping is preferred, in particular when the time delay between the start of each of said steps is selected to be $\pm k(ST)$, with k a natural number, S the size of the metric determined in such a step, T the time unit for determining one metric calculation. This restriction enables exchange of information, thus metric initialization, between these steps. An example of such scheduling of more than two compute and determining steps is shown in Fig. 12, showing 6 windows in a single combination and 3 sequential computations of valid state metrics. The grey dotted lines indicate metric reuses between the different windows. Such a combination of compute and determining steps is further denoted combined compute and determinining step, being characterised by two parameters. The first parameter M denotes the amount of windows, or compute and determining steps, in such a combination, while the second parameter K, denotes the amount of sequentially executed valid metric calculations. Note that the latency of such a combination can be expressed as (L+KS)T, with L referring to the dummy metric sequence.

[0054] In a further embodiment some particular combinations of these compute and determinining steps are selected based on optimization of both the latency of the resulting decoding method, the number of dummy sequences per output sequence and the memory size per dummy sequence.

[0055] In an aspect of the invention a combination with parameter K=2 is selected. Such a combination is graphically depicted in Fig. 11(a). Note that said method has one dummy sequence per output sequence and no memory reuse is exploited here.

[0056] In an aspect of the invention a combination with parameter K=3 is selected. Such combinations are graphically depicted in Fig. 13 and 11(b). The memory size per dummy sequence is here 0.5 and 1 dummy sequence per out-

BNSDOCID: <EP___1030457A2_I_>

put sequence.

[0057] In an aspect of the invention a combination with parameter K=4 is selected. Such combinations are graphically depicted in Fig. 14. The structure in Fig 14(a) has 0.5 dummy sequences for one output sequence. This structures minimize the number of computations of dummy metrics. The structures in Fig. 14(b) and (c) on the other hand show better memory reuse characteristics. Note that the grey shaded areas symbolize the memory space needed as function of time. Two memories for storage of state metrics while computing six output sequences is found for these last structures, at the expense of a little more dummy sequences per output sequence (2/3).

[0058] In an aspect of the invention a combination with parameters K=5 is selected. An example with M=16 is shown in Fig. 15.

[0059] Said third aspect of the invention may be carried out in an alternative way as represented in Fig. 16. The first state metrics ALPHA, being determined with a forward recursion, are divided in sets or classes (1)(2)(3). The second state metrics BETA, being determined with a backward recursion, are also divided in sets or classes (1)(2)(3). The first state metrics of one of such sets and the second state metrics of the corresponding set are computed both by a compute and determining step (4)(5). There are a plurality of such compute and determining steps. Observe in Fig. 16 that the compute and determining step (4) provides an input (6) to compute and determining step (5), more in particular for providing a valid initialization to step (5). Note that said compute and determining steps (4)(5) are being scheduled such that said steps are executed at least partially overlapping in time.

[0060] In a fourth aspect of the invention a method for iterative decoding is provided. Turbo decoding is characterized by performing a first decoding step followed by a second decoding step, and iteratively performing these two decoding steps. After loading or inputting the encoded signal, a first decoding step is performed, which produces or determines a first intermediate data element. The second decoding step exploites an interleaved version of said first intermediate data element in order to produce or determine a second intermediate data element. With interleaved version is meant that essential the same values are stored in said data element but the ordering within said data element, being a vector, is changed. After this said first decoding step is performed again, exploiting an interleaved ersion of said second intermediate data element.

[0061] As said first decoding step exploits data from said second decoding step and vice versa, these decoding steps are not performed simultaneously. Therefore when said first decoding step is executed by a first circuit and said second decoding step is performed by second circuit, only one of said circuit is executing its operation at the same time. Therefore no efficient use of the available hardware is obtained.

[0062] Fig. 4 shows a graphical represention of a classical turbo decoding setup. One recognises a first decoder, denoted DEC1 which performs a first decoding step and a second decoder, denoted DEC2, which performs a second decoding step. Feedbacks between said decoders are indicated. In these feedbacks blocks, denoted by π and π^{-1}, are shown, representing interleavers. Fig. 5 (a) shows how classical interleavers can be implemented. The first decoder (at the left), here a SISO decoder, produces outputs in a first ordering and stores these outputs in a first memory M1, big enough to store these outputs. The second decoder (at the left), here also a SISO decoder, consumes later inputs, obtained during a previous iteration from the first decoder and stored in a second memory M2. Said input consumption is done according to a second ordering, being substantially different from said first ordering.

[0063] In the present invention an iterative decoding procedure, as used in turbo decoding, wherein beter hardware efficient is obtained, is presented. The invented method is based on performing said first and second decoding step on the same hardware. As no simultaneous writing and reading is necessary, and because said decoupled steps cannot overlap in time, one memory of the size of a data element, is sufficient. This is represented in Fig. 5 (b) showing a single decoder, here a SISO decoder, but the invention is not limited thereto. In a first iteration, shown in Fig. 5 (b), leftside, this decoder performs a first single decoder step, and during this single decoder step, it reads and writes to the same memory in a particular order. In a second iteration, shown in Fig. 5 (b), rightside, the same decoder performs also a second single decoder step, being substantially different than said first single decoder step. During said second iteration the decoder reads and writes to the same memory as in the first iteration but another reading and writing ordering is used. Note that it is sufficient to have a memory with a single port here. A flexible address generator is exploited to enable interleaver functionality of said single-port memory.

[0064] In an embodiment of this aspect of the invention one recognizes that by considering said iterative performing said first decoding step and second decoding step together as a method step and applying said method step only on part of said encoded signal, then this method step will be smaller in size, thus needing smaller memories. The turbo decoding procedure can then be done by performing a plurality of such smaller size method steps, each comprising iteratively performing first and second decoding steps, substantially simultaneously. Each of such smaller size method steps then has a memory, being assigned to it, and used for storage of the intermediate data elements, produced by it. This is illustrated in Fig. 6, showing a set-up wherein a plurality of blocks, each comprising a decoder module and a memory being connected in a feedback arrangement. Each of said decoder modules is adapted such that it can perform a plurality of decoding steps. More in particular said plurality of decoding steps are performed iteratively. Thus in a further embodiment of this aspect of the invention one considers further iterative performance of a plurality of decod-

BNSDOCID: <EP___1030457A2_I_>

ing steps instead of said first decoding step and second decoding step only.

[0065] In another embodiment of this aspect of the invention one recognizes that by considering said iterative performing of said first decoding step and second decoding step together as a method step and applying each of said method steps on the complete encoded signal, one can decode a plurality of encoded signals simultaneously with the same hardware. This again results in an increase of hardware efficiency. Again this is illustrated in Fig. 6, showing a set-up wherein a plurality of blocks, each comprising of a decoder module and a memory being connected in a feedback arrangement. Each of said decoder modules is being adapted such that it can perform a plurality of decoding steps on an encoded signal. Each of said decoder is dealing with another encoded signal.

[0066] In a fifth aspect of the invention architectures, which implements the presented decoding methods described above, are presented.

[0067] In a first embodiment of this aspect of the invention an architecture, shown in Fig. 7, being closely related to the hardware efficiency aspect described above, is disclosed. Such an apparatus determines a decoded signal from an encoded signal, and is adapted for inputting said encoded signal and outputting said decoded signal. Said apparatus comprises of a plurality of parallel modules (4). Each of said modules is adapted for performing decoding.

[0068] In a further embodiment each of said modules comprises of a decoder module (6) with at least two inputs and at least two outputs. A feedback connection or loop is made between one of said module inputs and one of said module outputs but a memory (7), being assigned to said decoder module, is incorporated within said feedback connection.

[0069] In a still further embodiment each of said modules is adapted for performing an iterative decoding method on parts of said encoded signal, said iterative decoding method comprising iteratively performing a sequence of decoding steps, wherein each of said decoding steps exploits an interleaved version of the intermediate data elements, being produced by the preceding decoding step, said data elements being stored in and read from the memory, being assigned to the iterative decoding method.

[0070] In a further embodiment said memory in said feedback connection is a single-port memory.

[0071] In another embodiment of this aspect of the invention the apparatus determines a plurality of decoded signals from a plurality of encoded signals. Said apparatus comprises of a plurality of parallel modules (4). Each of said modules is adapted for performing decoding of one encoded signal.

[0072] It must be emphasized that with memories here is meant any kind of storage unit.

[0073] In a further embodiment an apparatus for turbo decoding as shown in Fig. 7, is adapted for inputting an input signal and outputting an output signal, said input signal being a turbo encoding signal, comprising a sequence of blocks, each of said blocks comprising of Nb samples, said apparatus comprising of a first module (1) with at least as input said input signal, further being adapted for determining reliability measures on said input signal; a second module (2), being adapted for demultiplexing the first module output in N second module outputs; a storage unit (3); a plurality of third modules (4), each of said third modules having at least a module input and a module output, each of said third modules being adapted for performing decoding of one of said blocks; and a fourth module (5), adapted for multiplexing the module outputs of said third modules, having as output said output signal. In said aparatus said reliability measures are log likelihood ratios.

[0074] Said apparatus for turbo decoding has an amount of third modules being an integer and
said storage unit is being adapted for inputting said N second module outputs and outputting M storage unit outputs, each of said M storage unit outputs being an input of a separate third module. Said storage unit comprises of N first memories (8) and a N to M switch (9), M being N-1 or wherein N being larger than or equal to M.

[0075] Said apparatus for turbo decoding is further characterized in that said decoding in said third modules comprises an iterative sequence of a set of ordered tasks. Each of said third modules comprises of a decoder module (6) with at least a first input, a second input, a first output and a second output, said decoder module being adapted for executing said ordered tasks one after another; a second memory (7) with at least a memory input and a memory output; and said first input being said third module input, said second input being said memory output, said first output being said third module output and said second output being said memory input.

[0076] In said apparatus said second memories are single port memories. Further a flexible address generator, being adapted for enabling interleaver functionality of said single port memories, is foreseen.

[0077] In a second embodiment of this aspect of the invention an architecture, being closely related to the parallisation aspect described above, is disclosed. Said apparatus determines a decoded signal by combining an encoded signal, first state metrics, being determined with a forward recursion, and second state metrics, being determined with a backward recursion. Said apparatus is adapted for scheduling a plurality of compute and determining steps, wherein each of said compute and determining steps comprises computing a part of said first state metrics with a forward recursion and a part of said second state metrics with a backward recursion, and determining part of said decoded signal by combining part of said encoded signal, part of said first state metric and part of said second state metric. Said apparatus schedules at least two of said compute and determining steps such that initialisation of one of said recursions for said computing of state metrics of one of said compute and determining steps is based on a computed value of said

BNSDOCID: <EP___1030457A2_I_>

recursion of said state metrics in the other compute and determining step.

[0078] In a further embodiment dedicated modules for each of said compute and determining steps are foreseen. Each of said modules is thus being adapted for executing one of said compute and determining steps. Moreover said apparatus schedules at least two of said compute and determining steps such that execution of said two steps being partially overlapping in time.

[0079] In a further embodiment said dedicated modules are sharing memory. More in particular the scheduling of said compute and determining step, executed by said modules, is done such that at least two of said modules share a memory for storage of state metrics, computed by said modules. Said shared memory is substantially smaller than two times the size of said state metrics.

[0080] In a third embodiment of this fifth aspect of the invention, architectures, being related to said recomputation methods, are presented. Said architectures are characterised in that a memory is assigned for storing part of said computed state metrics. The size of said memory is substantially smaller than the size needed for storing all said computed state metrics.

[0081] In a further embodiment a second memory for temporary storing recomputed state metrics is provided, the size of this second memory being such that it can at most store the amount of state metrics that need to be recomputed in between said stored state metrics.

[0082] In a fourth embodiment of this fifth aspect of the invention, architectures, being related to said double flow turbo decoding methods, are presented. Said architecture is characterised in that it is adapted for inputting said encoded signal, computing first state metrics with a forward recursion and second state metrics with a backward recursion, part of each of said state metrics being determined by a validly initialized recursion, the part of the state metrics not determined by a validly intialized recusion being determined by performing a plurality of restricted recursions, said restricted recursions only determining part of these state metrics, and determining said decoded signal by combining said encoded signal, said first state metric and said second state metric.

[0083] In a further embodiment the apparatus has at most two memories assigned for storage of valid state metrics being determined by said restriced recursions. The size of said memories is the size of the valid state metrics being determined by one of such a restricted recursion.

[0084] In a sixth aspect of the invention optimization of parameters of a turbo decoding method, being adapted such that extra parametric freedom is available, is presented. Indeed performance in terms of power or energy consumption, latency, memory requirements are dependent on the selected parameters. Said parameter or degrees of freedom are selected by optimizing a costfunction incorporating at least one performance criteria such as power or energy consumption, latency and memory requirements. It must be emphasized that said parametric optimized turbo decoding method can be implemented in a custom essentially digital device or in general purpose device or in a combination thereof. In particular embodiment while designing the turbo decoding implementing device, one performs optimization of parameters of a turbo decoding method, and selects optimal value of said parameters, thereby thus defining said selected parameters, and one designs said device dedicated for a method with said selected parameters. In an other embodiment one still performs optimization of parameters of a turbo decoding method, but one selects a range of optimal values. One designs a device which can perform said turbo decoding method for said range of values and for instance based on the available resources, e.g. memory parameters are selected within the optimal range. Further said parameter values can be changed on the fly, while executing the turbo decoding functionality of said device, based on a change of available resources, e.g. memory can be shared with other processes executed by said device or because the performance requirements changes as function of time.

[0085] In an embodiment of this sixth aspect of the invention one of said selected parameters is the parameter indicating how many state metrics are partially stored. This optimization refers to the partial storage and recomputation aspect of the invention.

[0086] In another embodiment of this sixth aspect of the invention one of said selected parameters is the parameter indicating the relative amount of dummy, invalid metrics per amount of valid metrics. This parameter is highly related to the decoding performance. Note that this parameter can be defined as the length of sequence of valid metrics divided by the length of the sequence of dummy metrics.

[0087] In yet another embodiment optimization of the parameter M denoting the amount of windows, or compute and determining steps in said parallelisation aspect of the invention is presented.

[0088] In an another embodiment optimization of the parameter K, denoting the amount of sequentially executed valid metric calculations, in said parallelisation aspect of the invention, is disclosed.

[0089] It must be emphasized that optimization of any combination of the above mentioned parameters can be done also.

[0090] The present invention includes the following methods:

[0091] A first aspect of a first method for determining a decoded signal from an encoded signal, said method comprising the steps of:

BNSDOCID: <EP___1030457A2_I_>

inputting said encoded signal;
computing first state metrics with a forward recursion and a second state metric with a backward recursion, and storing only a part of said computed first state metrics in a first memory;
determining said decoded signal from at least said encoded signal, said computed stored first state metric and said second state metric.

[0092] A second aspect of the first method, wherein said decoded signal determining step comprises further recomputing non-stored first state metrics, being intermediate said stored first state metrics, from said stored first state metrics, and said decoded signal determining step exploits said recomputed non-stored first state metrics.

[0093] A third aspect of the first method, wherein said decoded signal determining step comprises further recomputing and temporarily storing in a second memory, non-stored first state metrics, being intermediate said stored first state metrics, from said stored first state metrics and said temporary stored first state metrics, and said decoded signal determining step exploits said recomputed and temporary stored first state metrics.

[0094] The first, second or third aspects above of the first method, wherein said decoded signal determining step is performed while executing said backward recursion.

[0095] A fourth aspect of the first method, wherein said decoded signal determining step exploits a calculation step, wherein said decoded signal is determined directly from said encoded signal, said stored computed first state metric and said second state metric.

[0096] A second method for determining a decoded signal from an encoded signal, said method comprising an iterative decoding procedure comprising the steps of:

inputting said encoded signal;
performing a first decoding step for determining based on said encoded signal a first intermediate data element, being stored in a first memory;
performing a second decoding step for determining based on said encoded signal and an interleaved version of said first intermediate data element, read from said first memory, a second intermediate data element, being stored in said first memory;
iteratively performing said first decoding and second decoding step, wherein said first decoding step exploits an interleaved version of said second intermediate data element, being read from said first memory and stores said first intermediate data element in said first memory, said second decoding step exploits an interleaved version of said first intermediate data element, being read from said first memory and stores said second intermediate data element in said first memory; and
determining said decoded signal for either one of said intermediate data elements.

[0097] A third method for determining a decoded signal from an encoded signal, said method comprising the steps of:

inputting said encoded signal;
performing substantially simultaneously a plurality of iterative decoding methods on parts of said encoded signal, wherein each of said iterative decoding methods comprises:

performing a first decoding step for determining, based on part of said encoded signal, a first intermediate data element which is stored in a memory which is assigned to the iterative decoding method on said part of said encoded signal;
performing a second decoding step for determining, based on part of said encoded signal and an interleaved version of said first intermediate data element read from said memory, a second intermediate data element which stored in said memory;
iteratively performing said first decoding and second decoding steps, wherein said first decoding step exploits an interleaved version of said second intermediate data element read from said memory and stores said first intermediate data element in said memory, said second decoding step exploits an interleaved version of said first intermediate data element read from said memory and stores said second intermediate data element in said memory; and
determining part of said decoded signal for either one of said intermediate data element; and

assembling said decoded signal from said determined parts of said decoded signal.

[0098] A fourth method for determining a plurality of decoded signals from encoded signals, said method comprising the steps of:

BNSDOCID: <EP___1030457A2_I_>

inputting said encoded signals;

performing substantially simultaneously a plurality of iterative decoding methods,

wherein each of said iterative decoding methods decodes one of said encoded signals, and each of said iterative decoding methods comprises:

performing a first decoding step for determining a first intermediate data element which isstored in a memory which is assigned to the iterative decoding method;

performing a second decoding step for determining, based on at least an interleaved version of said first intermediate data element read from said memory, a second intermediate data element which is stored in said memory;

iteratively performing said first decoding and second decoding step, wherein said first decoding step exploits an interleaved version of said second intermediate data element read from said memory and stores said first intermediate data element in said memory, said second decoding step exploits an interleaved version of said first intermediate data element read from said memory and stores said second intermediate data element in said memory; and

determining a decoded signal for either one of said intermediate data element.

[0099] A first apparatus for determining a decoded signal from an encoded signal, said apparatus being adapted for inputting said encoded signal, computing first state metrics with a forward recursion and a second state metric with a backward recursion, within said apparatus a memory being assigned for storing part of said computed first state metrics, the size of said memory being substantially smaller than the size needed for storing all said computed first state metrics, said apparatus further being adapted for determining said decoded signal from at least said encoded signal, said computed first state metric and said second state metric.

[0100] The first apparatus being adapted for scheduling a step of recomputing non-stored first state metrics, being intermediate said stored first state metrics, from said stored first state metrics, and said decoded signal determining step exploits said recomputed non-stored first state metrics. This apparatus, wherein a second memory is assigned for temporary storing in a second memory recomputed non-stored first state metrics, being intermediate said stored first state metrics, from said stored first state metrics and said temporary stored first state metrics.

Claims

1. A method for determining a decoded signal from an encoded signal, said method comprising the steps of

   inputting said encoded signal;

   computing first state metrics with a forward recursion and second state metrics with a backward recursion whereby for each of said state metrics only a part (4) is determined by a validly initialized recursion (1)(2), the part of the state metrics not determined by a validly initialized recusion (5) being determined by performing a plurality of non-validly initialized converging recursionsand whereby each of these recursions is only determining a part of those state metrics not being determined by a validly initialized recursion (3);

   determining said decoded signal by combining said encoded signal, said first state metric and said second state metric. (Figure 21)

2. The method recited in claim 1, wherein said decoded signal determining step is performed while executing said validly initialized recursion or while executing said non-validly initialized recursions or wherein said decoded signal determining step is performed partly while executing part of said non-validly initialized recursion and partly while executing said validly initialized recursion..

3. The method recited in claim 1 or 2, wherein said validly initialized recursions are scheduled such that these are executed substantially simultaneously.

4. The method recited in claim 1, wherein at least two, three or four of said non-validly initialized recursions are scheduled such that these are executed partly simultaneously.

5. The method recited in any previous claim, wherein non-validly initialized recursions comprise the step of computing dummy state metrics.

6. A method for determining a decoded signal by combining an encoded signal, first state metrics, being determined with a forward recursion, and a second state metrics, being determined with a backward recursion, said method

BNSDOCID: <EP___1030457A2_I_>

comprising the steps of: executing a plurality of compute and determining steps (4)(5), wherein each of said compute and determining steps comprises computing part of said first state metrics with a forward recursion and part of said second state metrics with a backward recursion, and determining part of said decoded signal by combining part of said encoded signal, part of said first state metric and part of said second state metric; at least two of said compute and determining steps being scheduled such that initialisation (6) of one of said recursions for said computing state metrics of one of said compute and determining steps being based on a computed value of said recursion of said state metrics in the other compute and determining step. (Figure 16)

7. The method recited in claim 6, wherein said compute and determining steps comprise the step of computing dummy state metrics.

8. The method recited in claim 6, wherein said compute and determining steps comprise the step of storing either said part of said first state metrics or said part of said second state metrics.

9. The method recited in claim 6, wherein said compute and determining steps comprise the step of determining said part of said decoded signal while computing either said part of said first state metrics or said part of said second metrics.

10. The method recited in claim 6, wherein said compute and determining steps comprise the step of selecting one of said state metrics, storing said part of said selected state metrics while computing, and determining said part of said decoded signal while computing said part of said state metrics not selected.

11. The method recited in claim 6, wherein said compute and determining steps are scheduled such that said steps are executed substantially simultaneously.

12. The method recited in claim 6, wherein at least two of said compute and determining steps are scheduled such that execution of said two steps being partially overlapping in time.

13. The method recited in claim 12, wherein said two compute and determining steps exploit the same memory for storage of said selected state metric, the size of said memory being less than two times the size of said selected state metric.

14. A method for determining a decoded signal from an encoded signal, said method comprising the steps of:

inputting said encoded signal;
performing substantially simultaneously a plurality of iterative decoding methods on parts of said encoded signal, wherein each of said iterative decoding methods comprises:

iteratively performing a sequence of decoding steps, wherein each of said decoding steps exploits an interleaved version of the intermediate data elements being produced by the preceding decoding step, said data elements being stored in and read from a memory, which is assigned to the iterative decoding method on said part of said encoded signal; determining part of said decoded signal for one of said intermediate data elements; and

assembling said decoded signal from said determined parts of said decoded signal.

15. A method for determining a plurality of decoded signals from a plurality of encoded signal, said method comprising the steps of:

inputting said encoded signals;
performing substantially simultaneously a plurality of iterative decoding methods,
wherein each of said iterative decoding methods decodes one of said encoded signals, each of said iterative decoding methods comprises:

iteratively performing a sequence of decoding steps, wherein each of said decoding steps exploits an interleaved version of the intermediate data elements produced by the preceding decoding step, said data elements being stored in and read from a memory which is assigned to the iterative decoding method;
determining said decoded signal for one of said intermediate data elements.

BNSDOCID: <EP___1030457A2_I_>

16. An apparatus for determining a decoded signal from an encoded signal, said apparatus being adapted for inputting said encoded signal and outputting said decoded signal, said apparatus comprising of a plurality of parallel modules, each of said modules being adapted for performing decoding, wherein each of said modules comprises:

    a decoder module with at least a first input, a second input, a first output and a second output;
    a memory, being assigned to said decoder module; and
    a feedback connection between said second output and said second input; and said memory being incorporated within said feedback connection.

17. The apparatus recited in claim 16, wherein each of said modules is adapted for performing an iterative decoding method on parts of said encoded signal, said iterative decoding method comprising iteratively performing a sequence of decoding steps, wherein each of said decoding steps exploits an interleaved version of the intermediate data elements produced by the preceding decoding step, said data elements being stored in and read from the memory.

18. An apparatus for determining a plurality of decoded signals from a plurality of encoded signals, being adapted for inputting said encoded signals and outputting said decoded signals, said apparatus comprising of a plurality of parallel modules, each of said modules being adapted for performing decoding of an encoded signal, wherein each of said modules is adapted for performing an iterative decoding method on an encoded signal, said iterative decoding method comprising iteratively performing a sequence of decoding steps, wherein each of said decoding steps exploits an interleaved version of the intermediate data elements produced by the preceding decoding step, said data elements being stored in and read from the memory which assigned to the iterative decoding method.

19. An apparatus for determining a decoded signal by combining an encoded signal, first state metrics determined with a forward recursion, and second state metrics determined with a backward recursion, said apparatus being adapted for scheduling a plurality of compute and determining steps, wherein each of said compute and determining steps comprises computing a part of said first state metrics with a forward recursion and apart of said second state metrics with a backward recursion, and determining part of said decoded signal by combining a part of said encoded signal, a part of said first state metric and a part of said second state metric; at least two of said compute and determining steps being scheduled such that initialisation of one of said recursions for said computing of state metrics of one of said compute and determining steps being based on a computed value of said recursion of said state metrics in the other compute and determining step.

20. The apparatus recited in claim 19 wherein said apparatus comprises a plurality of modules, each of said modules being adapted for executing one of said compute and determining steps, and said apparatus schedules at least two of said compute and determining steps such that execution of said two steps is partially overlapping in time.

21. The apparatus recited in 19, wherein said apparatus comprises a plurality of modules, each of said modules being adapted for executing one of said compute and determining steps, and at least two of said modules share a memory for storage of state metrics, computed by said modules, said memory being substantially smaller than two times the size of said state metrics.

22. An apparatus for determining a decoded signal from an encoded signal, said apparatus being adapted for inputting said encoded signal, computing first state metrics with a forward recursion and second state metrics with a backward recursion, part of each of said state metrics being determined by a validly initialized recursion, the part of the state metrics not determined by a validly intialized recusion being determined by performing a plurality of restricted recursions, said restricted recursions only determining part of these state metrics, and determining said decoded signal by combining said encoded signal, said first state metric and said second state metric.

23. The apparatus recited in 22, having at most two memories assigned for storage of valid state metrics being determined by said restriced recursions and the size of said memories being the size of the valid state metrics being determined by such a restriced recursion.

24. An apparatus for determining a decoded signal from an encoded signal, said apparatus being adapted for executing a turbo decoding method, in particular in accordance with either 1, 6, 14 or 15, said turbo decoding method using at least one of the following parameters:

    size of the memory for partially storage of computed state metrics;

BNSDOCID: <EP___1030457A2_I_>

size of the memory for partially storage of recomputed state metrics;
the length of the sequence of valid metrics divided by the length of the sequence of invalid metrics;
the number of compute and determining steps;
the amount of sequentially executed valid metric calculations per compute and determining step;

characterised in that for at least one of said parameters either a range or said parameter value being selected by optimizing a costfunction incorporating at least one performance criteria such a energy consumption, latency and memory requirements.

BNSDOCID: <EP___1030457A2_I_>

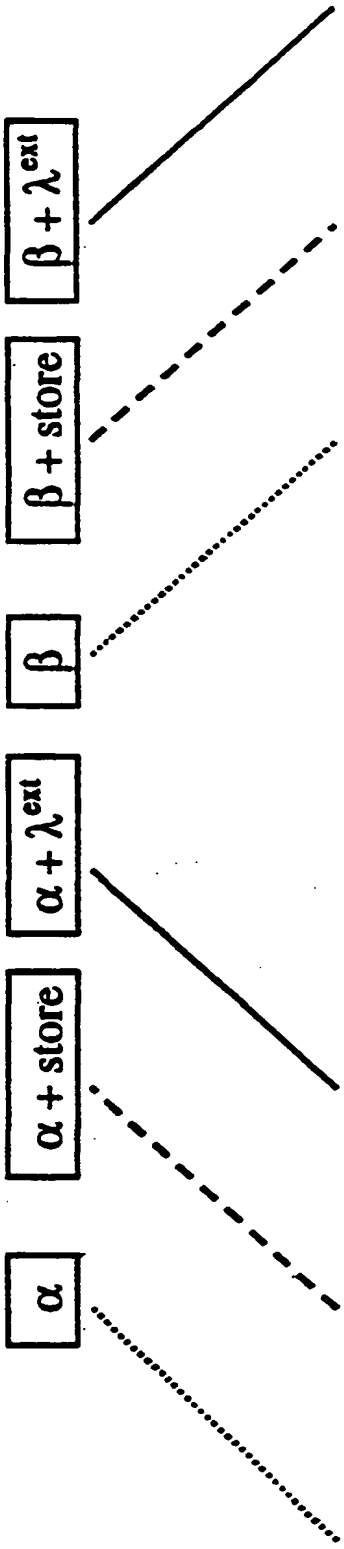


Figure 1

BNSDOCID: <EP___1030457A2_I_>

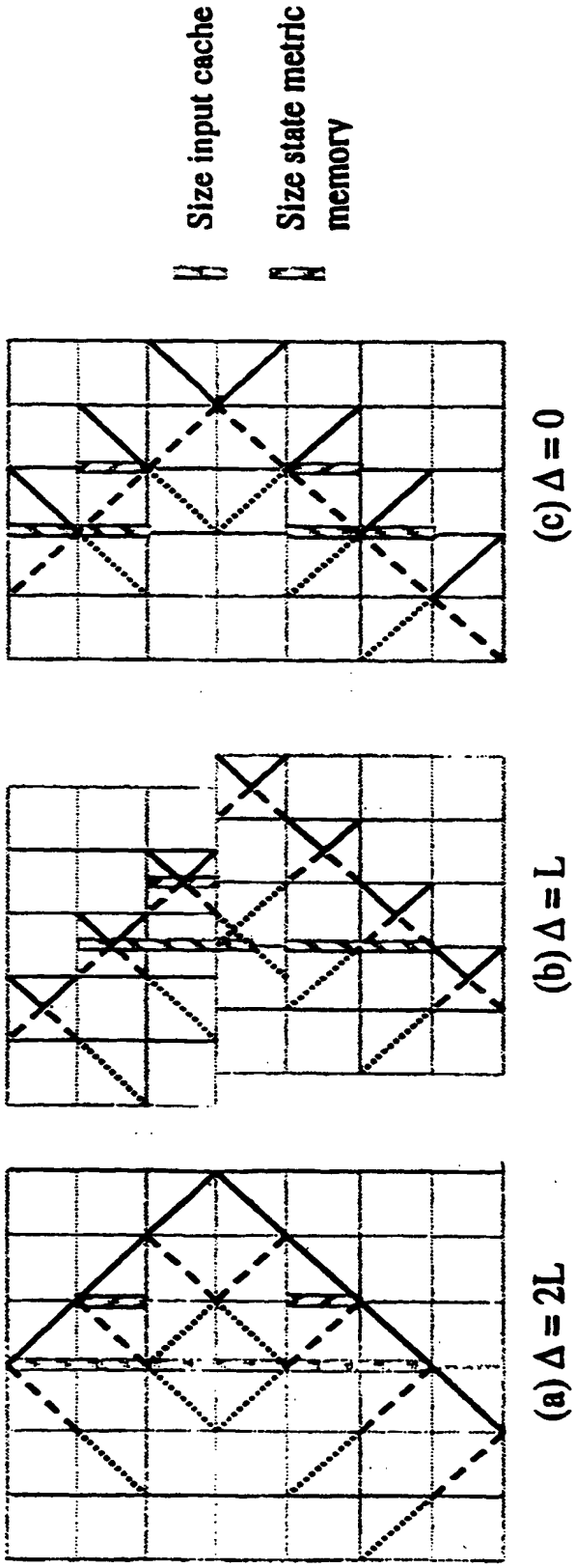


Figure 2

BNSDOCID: <EP___1030457A2_I_>

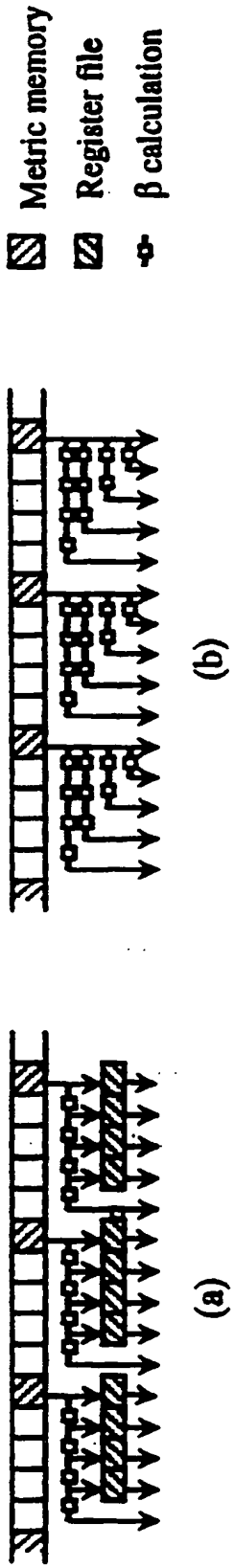


Figure 3

BNSDOCID: <EP___1030457A2_I_>

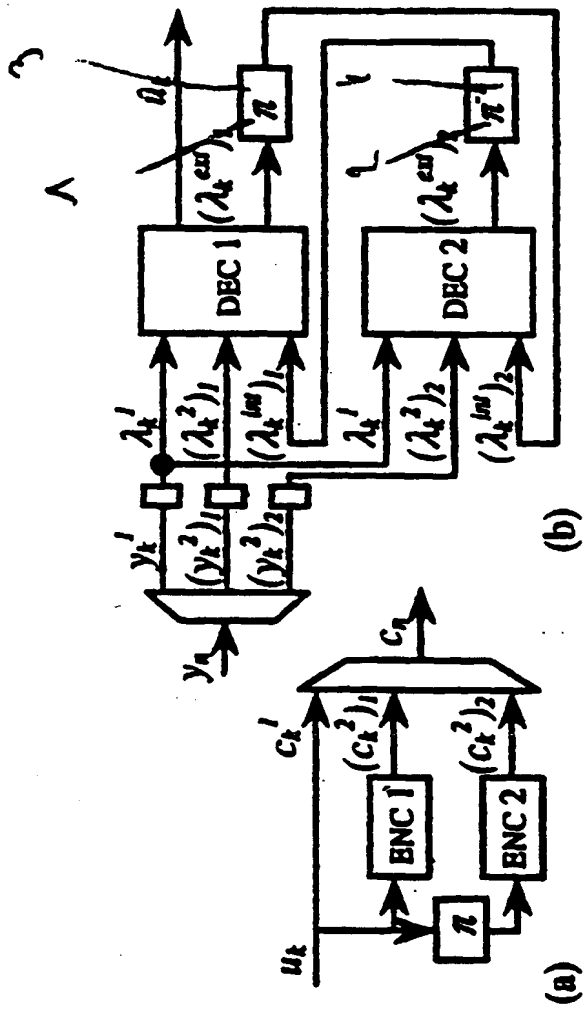


Figure 4

BNSDOCID: <EP___1030457A2_I_>

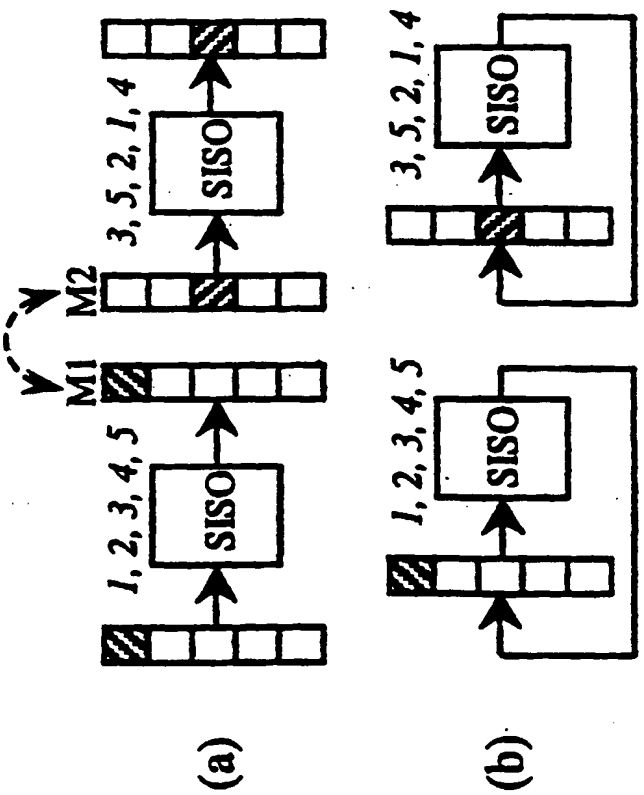


Figure 5

BNSDOCID: <EP___1030457A2_I_>

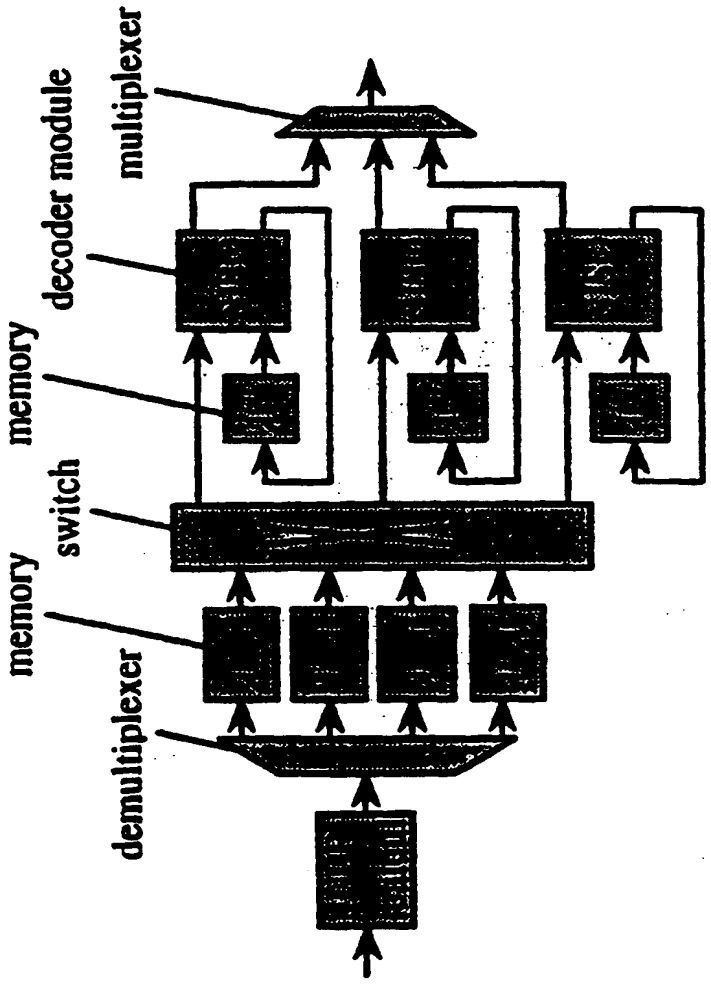


Figure 6

BNSDOCID: <EP___1030457A2_I_>

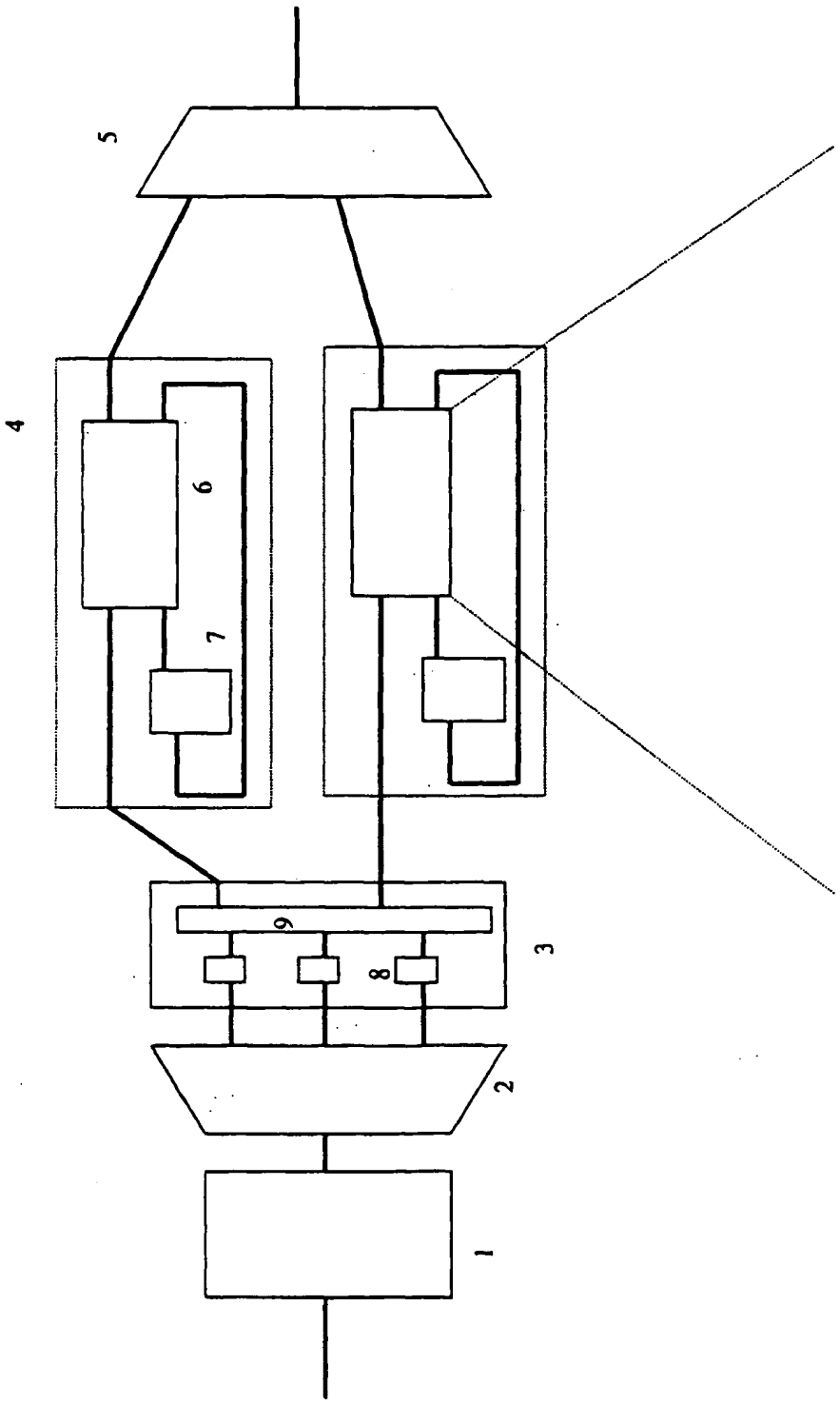


Figure 7

BNSDOCID: <EP___1030457A2_I_>

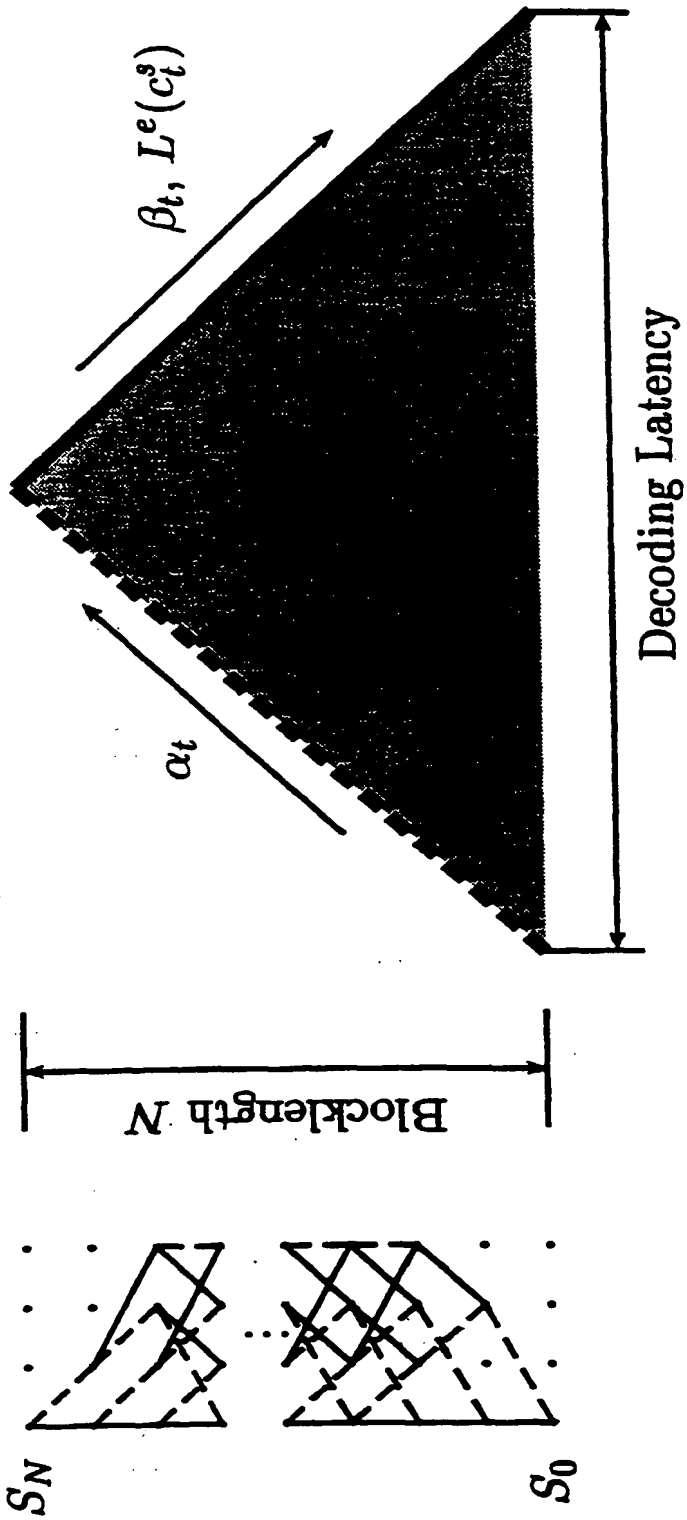


Figure 8

BNSDOCID: <EP___1030457A2_I_>

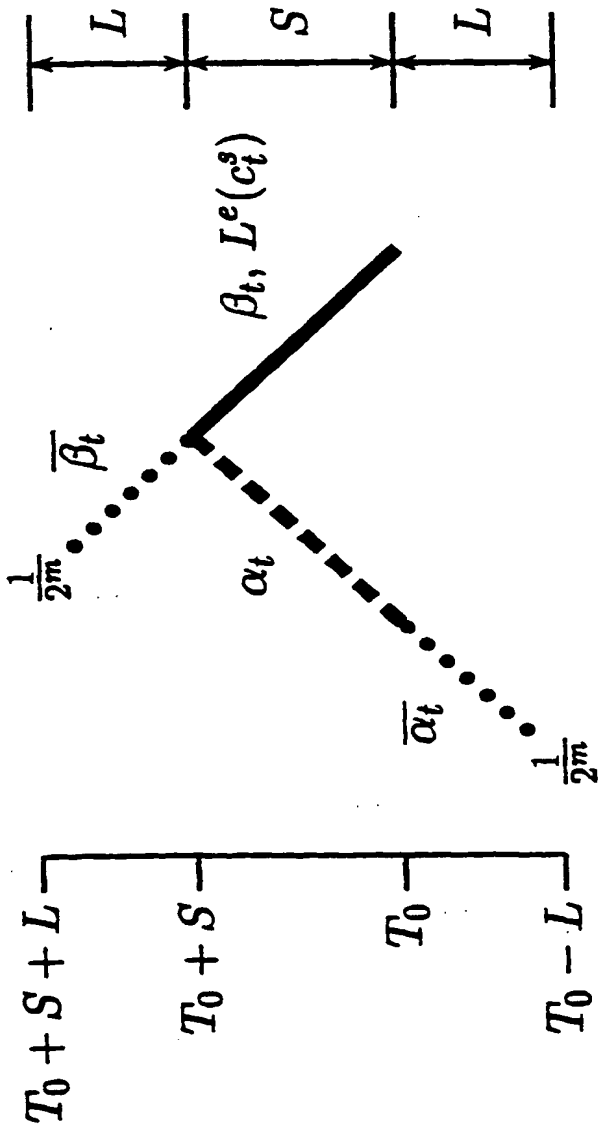
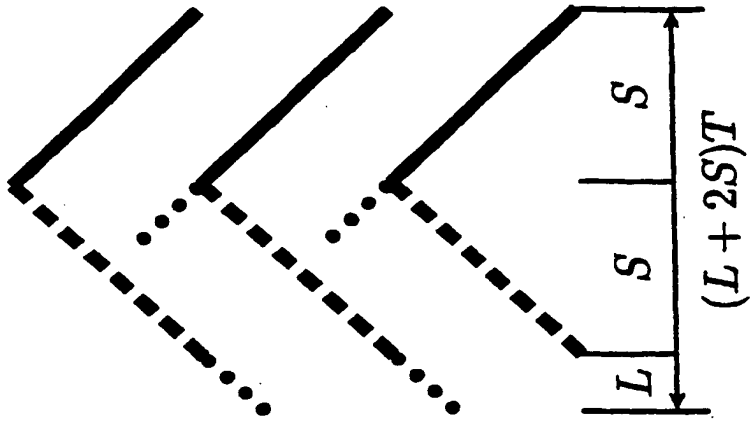
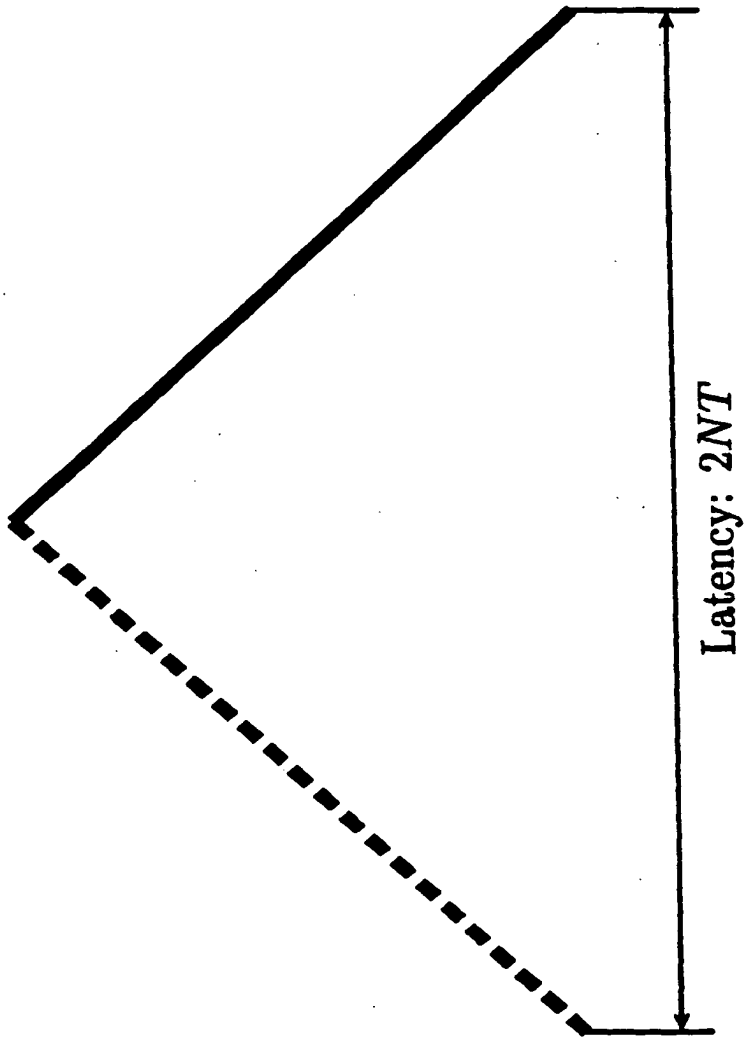


Figure 9

BNSDOCID: <EP___1030457A2_I_>



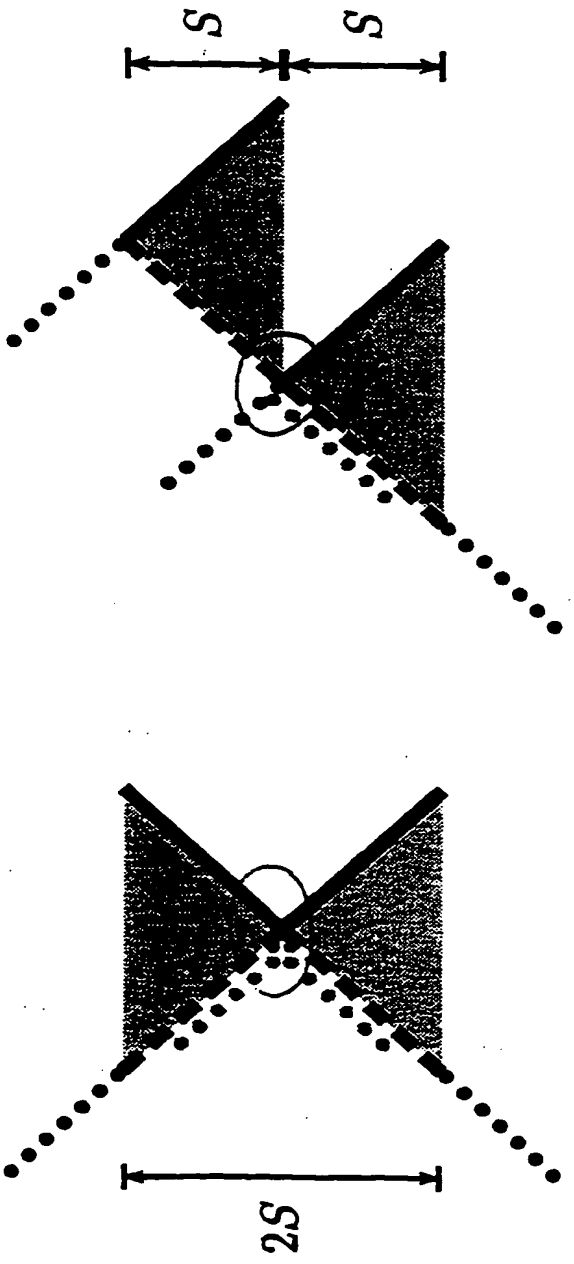
(b) The parallelized algorithm (3 windows)



(a) The standard algorithm

Figure 10

BNSDOCID: <EP___1030457A2_I_>



(b) Memory and metric reuse

(a) Metric reuse

Figure 11

BNSDOCID: <EP___1030457A2_I_>

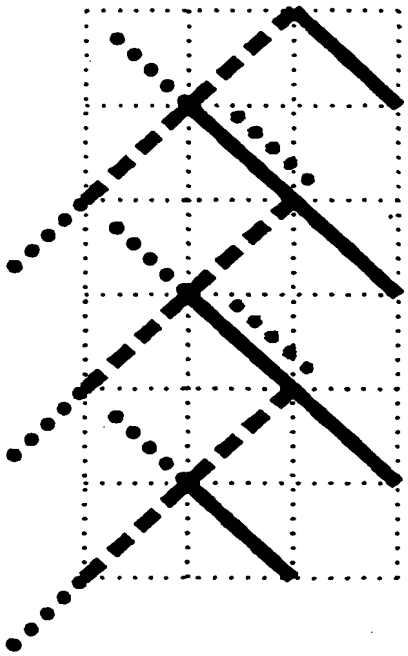
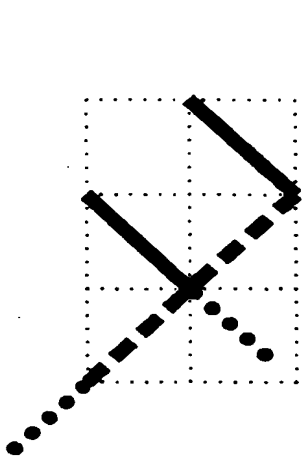
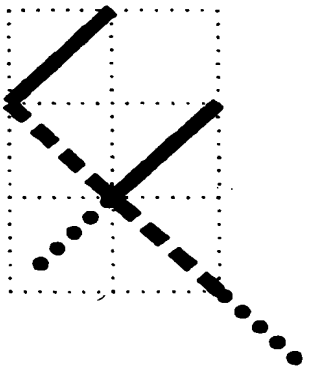


Figure 12

BNSDOCID: <EP___1030457A2_I_>



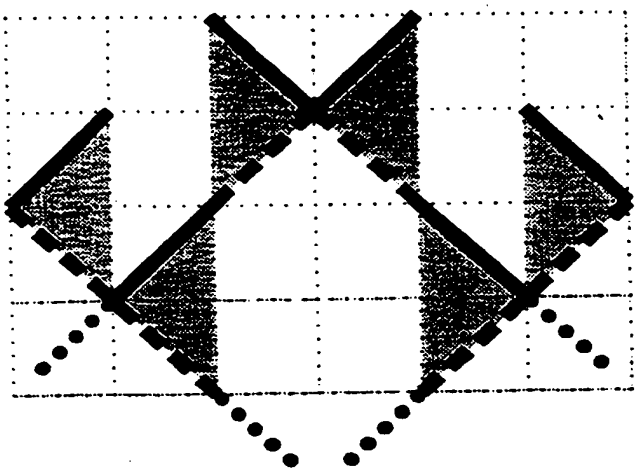
(a) $K = 3, \mu = \frac{1}{2}, \delta = 1$, type A



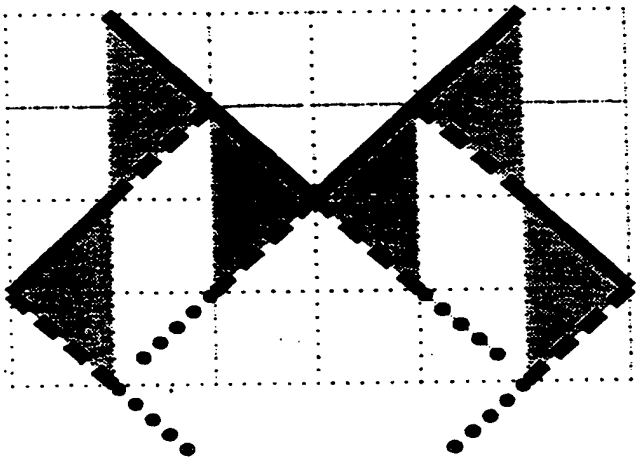
(b) $K = 3, \mu = \frac{1}{2}, \delta = 1$, type B

Figure 13

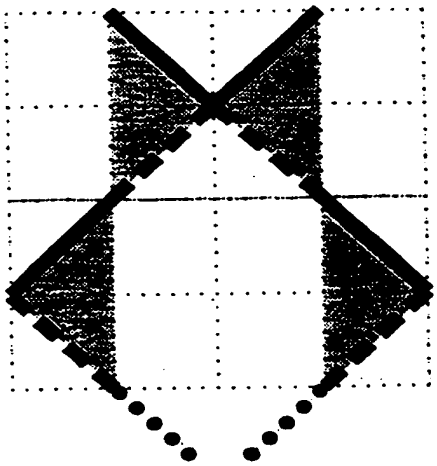
BNSDOCID: <EP___1030457A2_I_>



(c) $\mu = \frac{1}{3}, \delta = \frac{2}{3}$



(b) $\mu = \frac{1}{3}, \delta = \frac{2}{3}$



(a) $\mu = \frac{1}{2}, \delta = \frac{1}{2}$

Figure 14

BNSDOCID: <EP___1030457A2_I_>

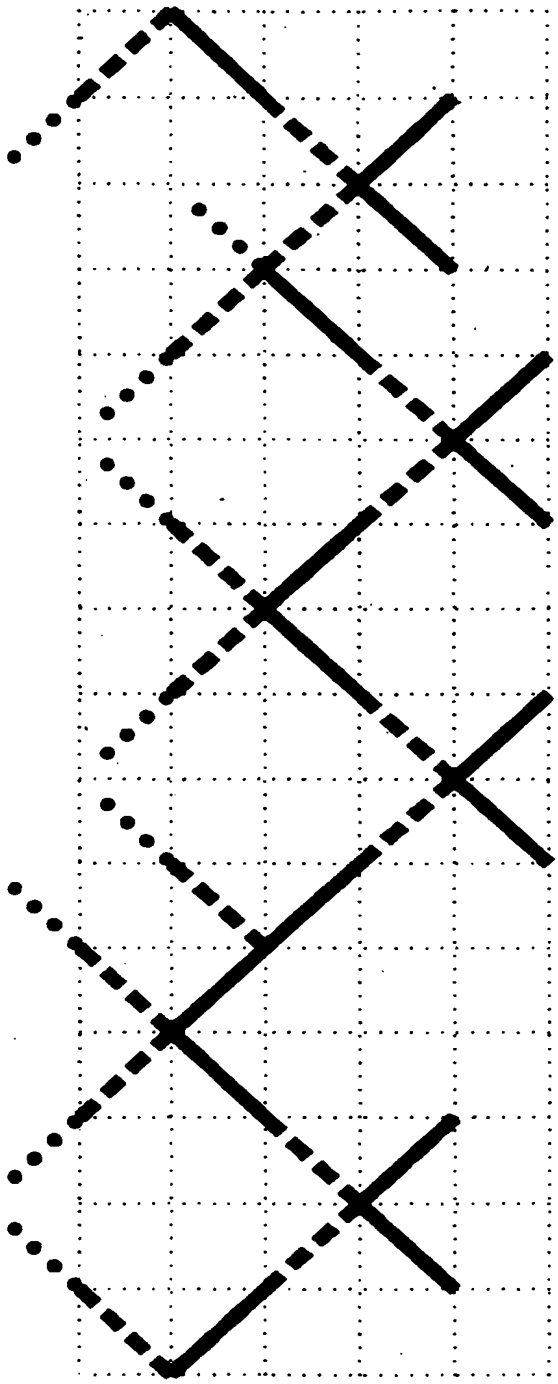


Figure 15

BNSDOCID: <EP___1030457A2_I_>

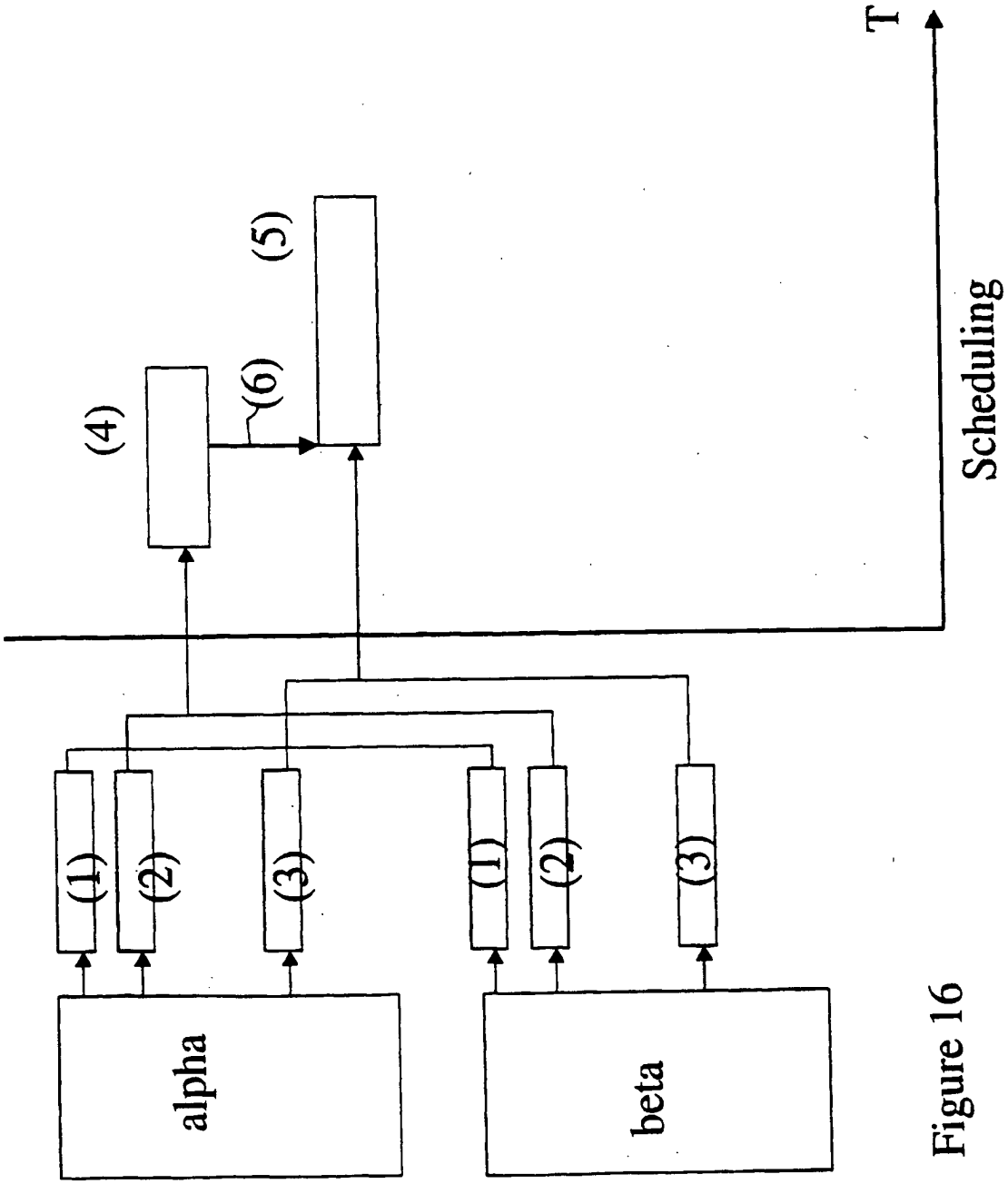


Figure 16

BNSDOCID: <EP___1030457A2_I_>

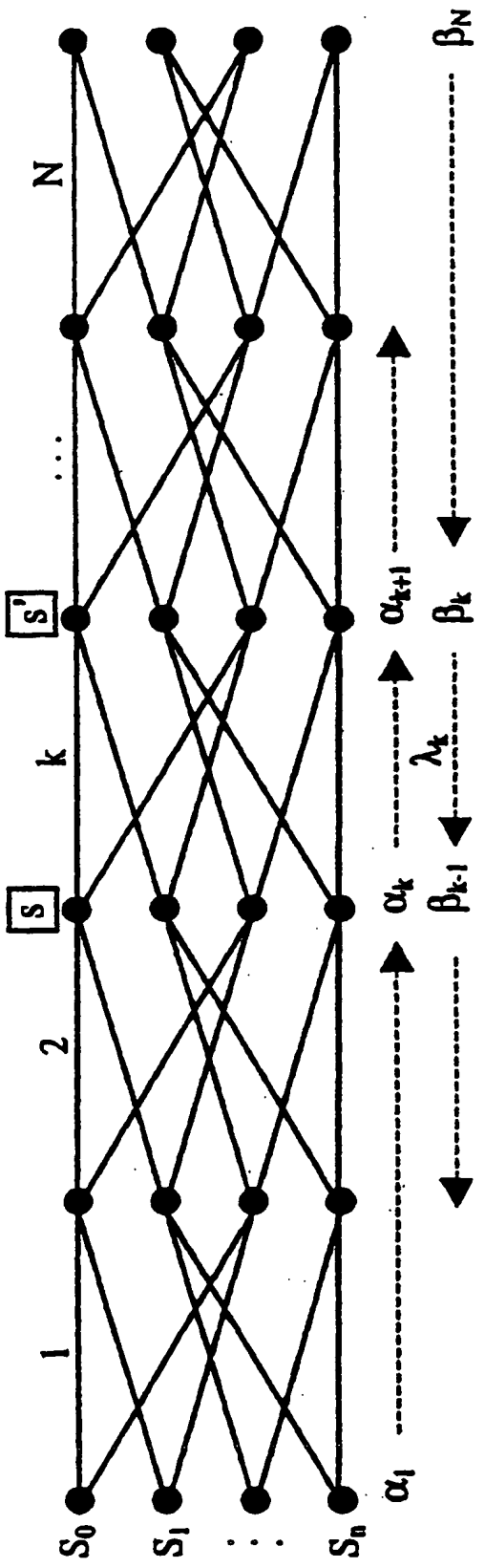


Figure 17

BNSDOCID: <EP___1030457A2_I_>

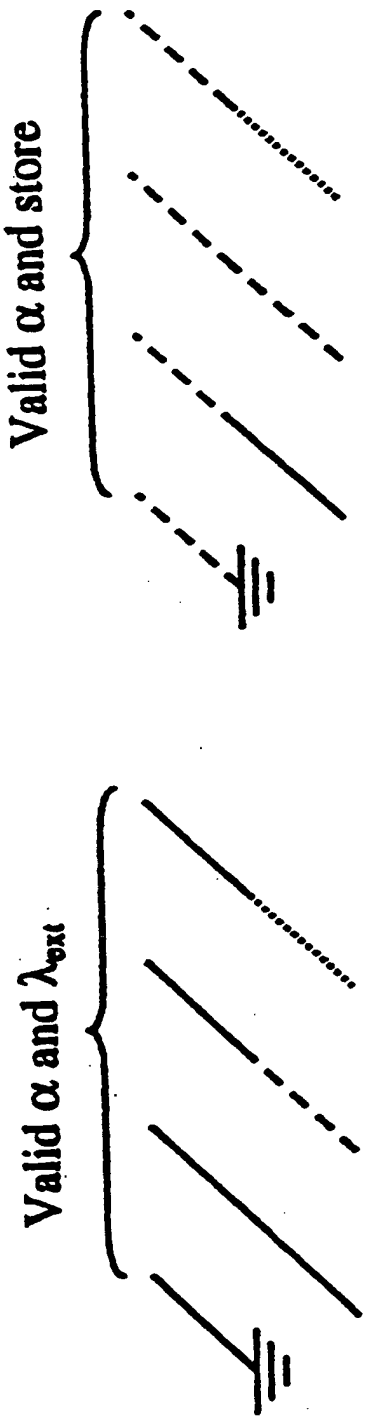


Figure 18

BNSDOCID: <EP___1030457A2_I_>

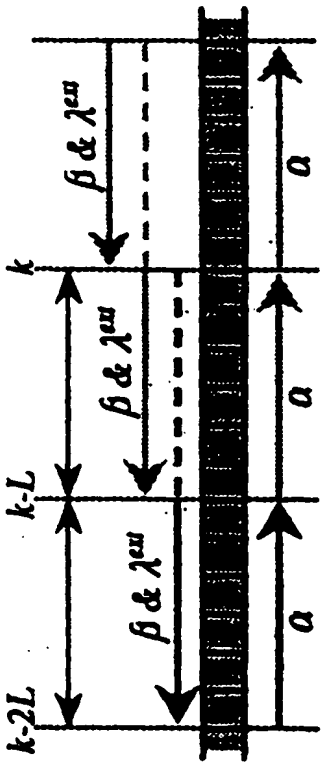


Figure 19

BNSDOCID: <EP___1030457A2_I_>

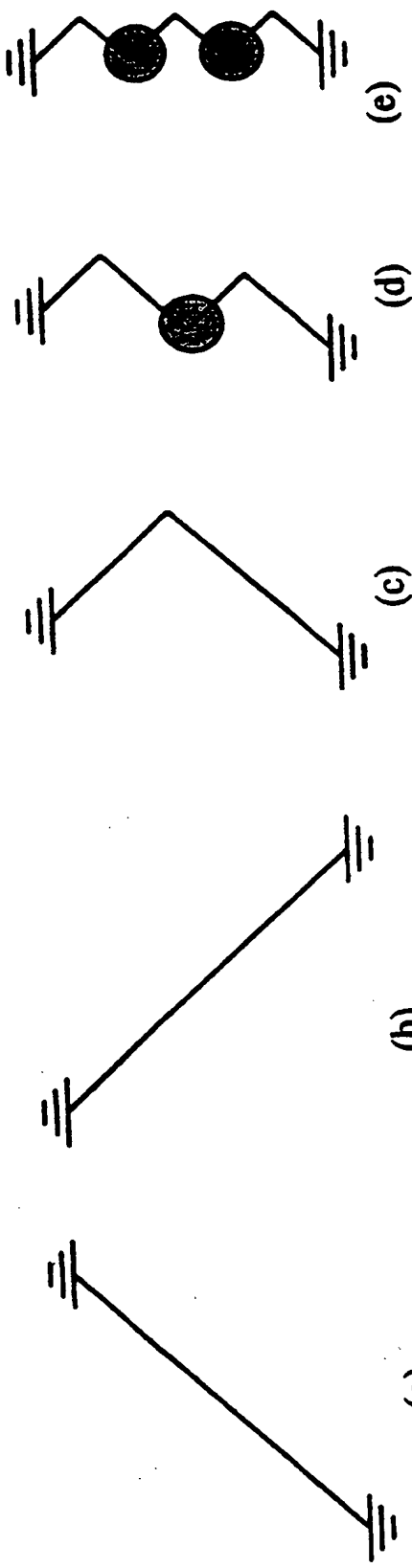


Figure 20

BNSDOCID: <EP___1030457A2_I_>

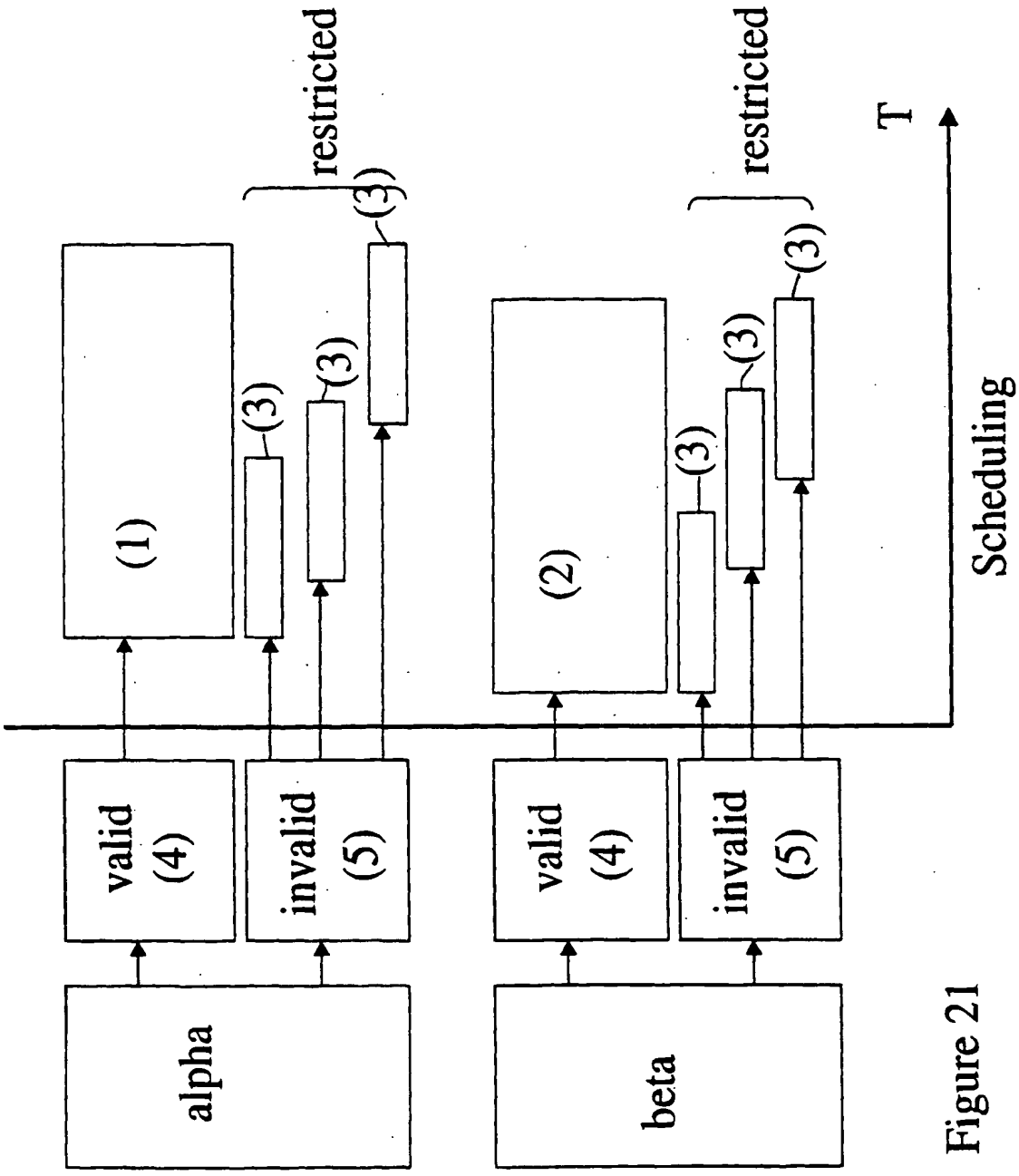


Figure 21

BNSDOCID: <EP___1030457A2_I_>

This Page Blank (uspto)